EnCompass Knowledge Systems, Inc.

Visualizing the Present, Designing the Future,
Managing the Transition[sm]

EnCompass®
Enterprise Metrics

17 September 2002

EXHIBIT A
Title: Business Analysis & Management Systems
Utilizing Enterprise Metrics
Inventors: Michael M. Mann & Arne Haugland
Attys.: Fulwider Patton et al. Dkt. # 65568/ENCMF

TABLE OF CONTENTS

LIST OF FIGURES

PROGRAM FLOW DIAGRAMS

EnCompass Enterprise Metrics

1.0 Introduction

EnCompass metrics form part of the EnCompass system, a computer-supported tool for enterprise process analysis and change management. The system does the following:

- collects data on the patterns of interaction among people in an organization
- analyzes these relationships with respect to specific organizational processes and issues
- presents the results in a graphical format

1.1 EnCompass Process Overview

The following broad outline characterizes the main steps in the EnCompass analysis process:

1. Definition of study issues and participants—EnCompass consultants meet with organization team members to determine the problems to be addressed and the specific *issues* that are likely to be relevant to the study (fig. 1).

Issue List - 7 Records

ID	Project	Type	Label	Description
ISO0000001	Sample DB	Issue 01	Product Creation	Product Development and Creation
ISO0000002	Sample DB	Issue 02	Mkt.Plng/Positioning	Market planning and positioning
ISO0000003	Sample DB	Issue 03	Cap.Plng and Utili.	Capital planning and utilization
ISO0000004	Sample DB	Issue 04	Order Fulfillment	Order Fulfillment
ISO0000005	Sample DB	Issue 05	Customer Comm.	Customer Communication
ISO0000006	Sample DB	Issue 06	Mgmt. Processes	Management Processes
ISO0000007	Sample DB	Issue 07	Fiscal Management	Fiscal Management

Figure 1: Project issues

2. Data collection—EnCompass creates a *data collection instrument* (fig. 2) to gather information on organization members' interactions with each other. Study participants indicate the specific individuals they interact with on a regular basis, and rate the overall frequency and importance of those interactions and their impact on each of the specific issues that were identified in step 1, on a scale of 1 to 5. The exact meanings of these values may vary from study to study; typical ranges are from "less than once a month" (1) to "several times a day" (5) for frequency, "not important" (1) to "critically important" (5) for importance, and "seldom" (1) to "always" for impact. (Zero represents no interaction.) In addition to the frequency, importance, and impact values, survey respondents may also be asked to estimate the number of hours per week spent per week interacting with each person on the list, and report the primary *modus* of these interactions: whether they are usually *concurrent* (face-to-face meetings, telephone, etc.) or *nonconcurrent* (email, memos, etc.).

Name John Frees ID#: 105

With whom do you interact at least monthly in ways that are generally important in order to get your tasks done, or to help the other person get his/her tasks done?

Frequency Range

Importance Range

IMPACT

Impact Scale Ranges 1 through 5 (0 "Never" leave blank)
Seldom 1 Occasionally 2 Half the time 3 Usually 4 Always 5

		Frequency	Importance	Product Creation	Mkt Plng/ Positioning	Capital Plng/ Utilization	Order Fulfillment	Cust Comm	Mgmt Proc	Fiscal Mgmt			
D.	Scales	4	4	4	5	0	3	5	4	4			
D	Vogel	3	3	2	3	4	2	2	2	2			
P.	Kilroy	3	4	0	0	1	0	0	2	2			
P.	Baird	2	3	2	2	0	1	1	0	0			
N.	Dooley	1	4	0	0	0	0	2	2	0			
J	Swardlow	2	3	1	1	3	1	1	2	1			
P	Packard	3	3	0	2	1	3	1	2	2			
C	Dinn	3	3	3	3	0	1	1	2	2			

Figure 2: EnCompass data collection instrument

Each line in the completed data collection instrument represents a regular interaction between the survey respondent and one other specific organization member. For each regular interaction, the respondent has estimated its overall frequency and importance and its impact on each of the identified study issues. When the survey information is entered into the EnCompass database, each line becomes a single *data collection record* (figs. 3, 4).

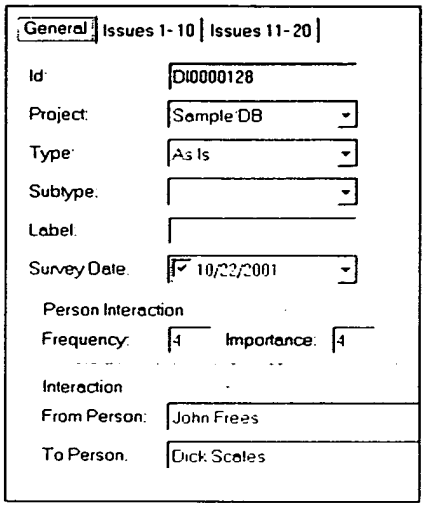

Figure 3: Data collection record—General data

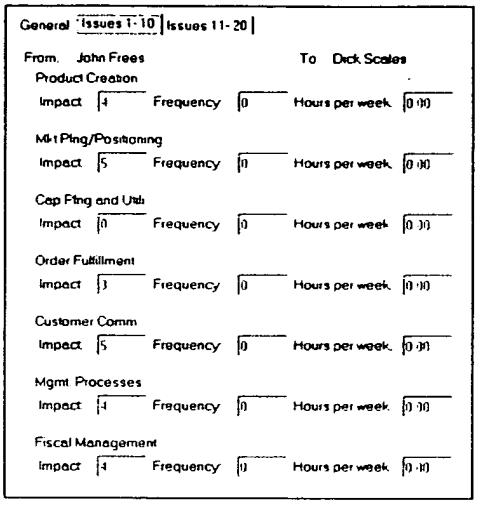

Figure 4: Data collection record—Issue-specific data

3. Data analysis—Using the EnCompass software, the consultant creates *analyses*, examining the interaction patterns of selected combinations of people and issues. For each analysis, the software converts the data to a three-dimensional graphical display (fig. 5, page 4). By interpreting the display and creating additional analyses to examine specific types of interaction, the analyst locates areas where decision-makers are not well integrated into the organizational structure, or where mismatched perceptions and inefficient communications channels may be interfering with the accomplishment of the organization's process objectives.

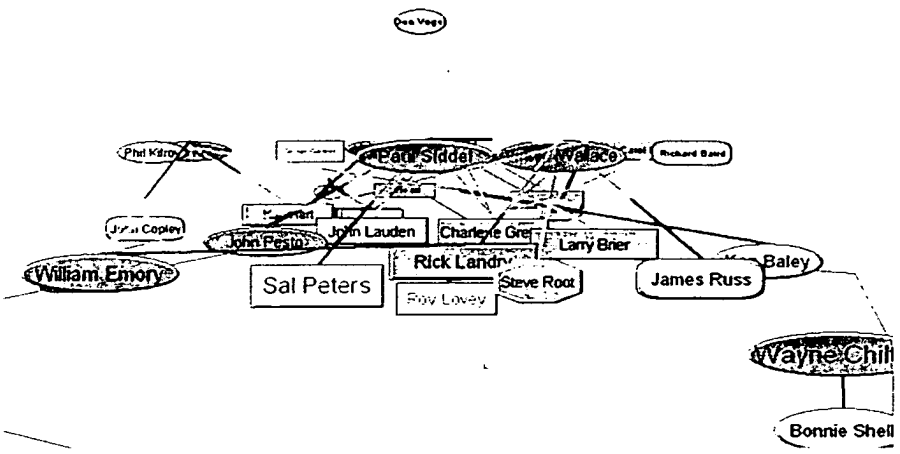

Figure 5: Example analysis display

4. Report and recommendations—EnCompass presents its findings, in the form of graphical reports and face-to-face meetings, and as an ongoing subscription service.

1.2 ENTERPRISE METRICS

Enterprise metrics are a set of tools available to the consultant in the data analysis stage of the process, through the EnCompass software. Using standard statistical methods, they measure the the amount of agreement among the organization's members with regard to the importance and impact of all their individual interactions with one another.

Taken singly, each of these indicators provides a means of quantifying an organizational characteristic such as "integration" or "clarity," which in other analytical settings is typically described in less precise and more impressionistic terms. Collectively, they amount to an overall "scorecard" of the degree to which an organization's communications and decision-making processes are integrated and effective. Once established, these figures can serve as benchmarks for orienting and monitoring organizational change and improvement.

1.3 SCOPE OF DISCUSSION

Each of the EnCompass enterprise metrics is based on a statistical equation that is automatically applied to a group of records, selected from an existing database

according to criteria previously defined in the EnCompass software. The context within which each of the results is displayed is likewise part of a previously existing system and is outside the scope of the present discussion. What is described here are:

- The statistical measurements themselves, as they are applied to the study of organizational dynamics
- The procedures by which the software selects the appropriate database records and computes the resulting values for each of the metrics

2.0 ENTERPRISE METRICS ANALYSIS

The remaining discussion presents the details of the EnCompass metrics analysis. Section 2.1 provides definitions of necessary concepts. Section 2.2 (page 8) describes the various metrics in detail. Section 2.3 (page 13) defines the data objects used in the program flow. Sections 2.4–2.6 present pictorial diagrams and detailed explanations of the program flow. Section 3 (page 52) dicusses options for visual output.

2.1 KEY CONCEPTS AND DEFINITIONS

Our description of the EnCompass enterprise metrics analysis employs the following concepts:

The **project** (or "study") is the large-scale unit of organization for work in EnCompass, typically involving the analysis of interactions among members of a single company or other organization. For each project, the project designer identifies the relevant *issues* to be examined, and creates a data collection instrument to measure interactions based on those quantities.

Issues are the concerns that drive decisions and activities within an organization, typically involving business processes such as "market planning" or "customer communication." For each study, the EnCompass consulting team works with representatives from the organization to identify the specific issues that need to be examined. The EnCompass data collection instrument measures the impact of

interactions between individuals with respect to every one of the identified issues; these values are stored in individual *data collection records*.

Data collection records contain information on the regular interactions between members of an organization in an EnCompass study, as gathered by the study's survey questionnaire. Each data collection record contains the rankings that a specific person (the "From" person) gave to his or her interaction with one other person (the "To" person), in terms of its overall **frequency** and **importance,** and its **impact** on each of the *issues* that the project designer has defined as significant (see figures 3 and 4, page 3). Valid values for frequency, importance, and impact are integers from 0 to 5.

A ***data collection link*** is a connection between a survey respondent (the "From" person) and another member of the organization (the "To" person), representing a regular interaction which the respondent judges to have some importance to his ability to accomplish his work-related tasks.

- A **confirmed** link is one that goes in both directions; that is, the interaction is reported by both parties. Every confirmed link, therefore, consists of two separate data collection records: one reflecting the interaction A → B, the other B → A. (Person A and person B may, however, make very different estimates of the interaction's importance, frequency, and impact.) An **unconfirmed** link is one that is reported by only one of the parties: A → B but not B → A. A **half-link** refers to the data collection record represented by either side of a confirmed link: A → B (the left side) or B → A (the right side).
- Confirmed links may be either ***agreed*** or ***nonagreed.*** An *agreed* link is one in which the values for importance, frequency, and impact as reported by both persons match each other, within a range of values specified by the agreement *criteria.* A high proportion of nonagreed links within an organization indicates a lack of effective communication about its processes.
- The EnCompass display engine converts these links to a graphical form for display and interpretation (see figure 5, page 4). Each link appears on a two- or three-dimensional tree diagram as a line between two nodes, representing the two parties to the interaction. The width and color of the lines indicate frequency,

importance, and issue-impact values. The analyst chooses which links to display by selecting one of the user-defined analysis definitions.

The ***analysis definition*** contains detailed information on the parameters that determine which *data collection links* are included in a given analysis. Each analysis selects *data collection records* from the database according to a number of *SQL queries*, which the EnCompass software creates automatically from choices made in the user interface.

SQL queries are statements which request specific information from a database, using the standard *Structured Query Language*. For example, the following SQL query:

> SELECT last_name, first_name FROM employees WHERE age < 50 ORDER BY last_name

would retrieve the last name and first name of all employees whose age is less than 50, sorting the output by last name.

Each EnCompass analysis selects *data collection records* for display based on the following types of user-defined queries, distinguished by the database location of the information they are searching for:

- **Data collection** queries look for information contained in the data collection records themselves, such as interaction frequency, importance, and impact. At least one data collection query is required for every analysis.
- **Person** queries select records based on information in the individual person records, such as corporate title, function, salary, or tenure. Person queries are optional when setting up an analysis.
- **Organization** queries search for records based on information about the organization, such as address or other defined attributes. Organization queries are optional.
- **Criteria** are ranges of values within which the two parties to an interaction must agree or disagree in order for the data collection record to be included in the current analysis. Criteria queries typically contain complex combinations of parameters. For example, an analyst might wish to view only those interactions where the impact values agree within a range of ±2 on the issue being examined, and within a range of ±5 on all other issues.

Model 1 and model 2: An analysis definition may contain a second complete set of search parameters, representing a second set of *SQL queries.* The database search engine retrieves the data collection records matching both of these query sets, and the analyst can elect to add the results together, subtract one from the other, or merge the results according to specific conditions.

> NOTE: In this discussion, the number of alternative analysis models is limited to two, because the existing EnCompass software only makes use of this number. The data analysis procedure itself, however, can incorporate additional models, if a suitable user interface is constructed to accommodate them.

2.2 ENTERPRISE METRICS DESCRIPTIONS

EnCompass enterprise metrics comprise six measurement groups. Section 2.2.1 provides brief descriptions of each; section 2.2.2 presents detailed descriptions of the computations for arriving at each measurement.

2.2.1 Brief descriptions

Clarity (or "alignment") is the overall level of agreement and understanding among the members of the organization or organizational unit.

Involvement is a measure of how effectively the organization's members are engaged in carrying out the tasks that are viewed as important to it.

Leverage is the degree to which the existence of an organization provides its members with greater influence than they would have as independent individuals.

Priority is a measure of the perceived impact of all the issues examined in an EnCompass study.

Relative Priority is a measure of the perceived impact of a selected issue, as compared with that of the other issues under consideration.

Integration measures the degree of interconnection between two organizations or

Int gration measures the degree of interconnection between two organizations or organizational units, by examining the links between them.

2.2.2 Detailed descriptions

This section contains a detailed description of each enterprise metric and the formula used to measure it.

2.2.2.1 Clarity

Clarity (or "alignment") refers to the overall level of agreement and understanding among the members of the organization or organizational unit. It is arrived at by summing the number of *confirmed links* over the total number of links, and can be either general or issue-specific, depending on the choice of analysis criteria.

$$\text{Clarity} = \frac{\text{Links(confirmed)}}{\text{Links(confirmed)} + \text{Links(unconfirmed)}}$$

The range of clarity is $0 \leq 1$, where 0 represents a total lack of clarity and 1 represents perfect agreement (within the preset agreement criteria).

Note that this metric includes only those individuals with links. *Isolates*, or individuals with no links satisfying the current analysis criteria, are excluded from the population in the denominator used for normalization. Therefore, all types of clarity depend on which issues are included, as well as what other selections are made, in any given analysis.

2.2.2.2 Involvement

Involvement is a measure of how effectively the organization's members are engaged in carrying out the tasks that are viewed as important to it. In general, it measures the number of confirmed links with *higher importance* against the total links. Specifically, confirmed links having EnCompass *Importance* values of 3, 4, or 5 (representing *important, very important,* and *critical,* respectively) represent effective involvement of the individuals involved. Links with importance of 2 or 1 (*moderate* and *minimal* importance), or isolated individuals with no links, represent limited involvement. As in the case of clarity, the Involvement parameter can be limited by the issues, criteria, or

attributes specified in the analysis definition, or be general (no other limiting criteria employed).

$$\text{Involvement} = \frac{L}{N(2^{N-1}-1)}$$

where: L = confirmed links with *Importance* ≥ 3

N = total population ($[2^{N-1}-1]$ represents the maximum number of links in a population of size N)

The range of involvement is $0 \leq 1$, where 0 = no important interactions with others and 1 = full involvement.

2.2.2.3 Leverage

Leverage is the degree to which the existence of an organization provides its members with greater influence than they would have as independent individuals. It is the sum of the confirmed links, weighted by their Importance values. Leverage can be issue- or attribute-dependent or independent, depending on the selection of the other criteria in the analysis.

$$\text{Leverage} = \frac{L_1 + 2L_2 + 3L_3 + 4L_4 + 5L_5}{5N(2^{N-1}-1)}$$

where: L_a = number of confirmed links with Importance = a

N = total population ($[2^{N-1}-1]$ represents the maximum number of links in a population of size N)

The range of leverage is $0 \leq 1$, where 0 = no leverage and 1 = maximum leverage.

2.2.2.4 Priority

Priority is a measure of the perceived impact of all the issues examined in an EnCompass study. It is the sum of the number of half-links, weighted by their impact values, and normalized by the population size. It is issue-dependent.

$$\text{Priority} = \frac{L_1 + 2L_2 + 3L_3 + 4L_4 + 5L_5}{10N(2^{N-1} - 1)}$$

where: L_a = number of half-links with Impact = a

N = total population ($[2^{N-1} - 1]$ represents the maximum number of links in a population of size N)

The range of priority values is $0 \leq 1$.

2.2.2.5 Relative Priority

Relative Priority measures the perceived impact of a selected issue as compared with that of the other issues under consideration.

$$\text{Relative Priority} = \frac{P_n}{\sum_i P_i}$$

where: Pn = Priority value of issue n

i = issue number

2.2.2.6 Integration

Integration measures the degree of interconnection between two separate organizations or organizational units, by examining the confirmed links between them. The analyst may define an "organizational unit" in any desired way, by selecting parameters in the Model 1 and Model 2 analysis definition panels.

The integration metric sums the number of confirmed links between the two units, weighted by their importance and normalized by the weighted product of the two populations (the total number of possible links). Integration can be either issue- or attribute-specific or -nonspecific.

$$\textbf{Integration} = \frac{L_1 + 2L_2 + 3L_3 + 4L_4 + 5L_5}{5N_1N_2}$$

where: L_a = number of confirmed links between unit 1 and unit 2 with Importance = a

N_1, N_2 = total number of links in unit 1 and unit 2

The range of integration is $0 \leq 1$, where 0 = no connection between units and 1 = full integration.

2.2.3 Scope, depth, and direction of metrics calculations

The metrics evaluate links between nodes on a tree structure. Therefore, the numbers they return depend on how many nodes are included in the calculation, and which ones. The program calculates the figures for every node (including the root) on the basis of a particular tree structure, whose depth and direction can be altered in two different ways. Noninteractively—that is, outside the user's control—the program can present **metrics families** which define various groups of nodes. Under the user's control within the user interface, the depth and direction of the tree display itself can limit the number of nodes to be included in the tree, and hence the calculation.

2.2.3.1 Choice of root node

The most general use of the EnCompass metrics is to examine the performance of an organization as a whole. For this, the analyst simply displays a tree with the highest-level entity as its root, and reads the figures attached to that node. However, metrics are in fact calculated and displayed for every node in the tree—that is, every person who is represented in a given analysis. When an analyst wishes to focus attention on a specific part of an organization, he or she can read the figures for any individual node.

2.2.3.2 Metrics families

It is possible to define *families* of metrics, which specify what group of nodes the calculations will be based on. Currently, there are two such families for all metrics. The **immediate** family includes only the node and its direct child nodes. The **extended** family includes a node and all those beneath it on the tree. Therefore, **immediat** metrics are based on a person's interactions with only those who report to him or her directly, whereas **extended** metrics produce figures reflecting his or her relationships with all subordinates at any level.

For the Clarity metric, there is an additional type: **node clarity** calculates the numbers of confirmed vs. unconfirmed links from the point of view a single node. In other words, the node's family includes only itself.

Other metrics families are also possible: for example, **influence** metrics count those persons anywhere in the organization who are within a person's *sphere of influence,* that is, those with whom the person has direct interaction links. **Organization** metrics measure any person's clarity, involvement, or other parameter with respect to the organization as a whole.

2.2.3.3 Depth and direction of the tree display

The user interface permits the analyst to limit the tree display to a desired number of levels. If the selected depth is less than that of the organization structure, the metrics calculations will change accordingly.

For any node, the *direction* of the tree also strongly affects the calculations for all nodes. It is possible to construct a tree upward through the data structure, so that the values for any individual node are calculated with respect to those above him or her in the organizational hierarchy, either direct superiors (in the case of the *immediate* metrics family) or all the way to the top of the organization structure (for *extended* metrics).

2.3 DATA OBJECT DEFINITIONS

In order to transform raw data collection records into links between persons in an organizational tree structure, the metrics analysis creates and manipulates a number of data objects (classes), consisting of pointers to columns in database tables. Some of these objects correspond closely to the structure of database records. Through a series of comparisons, these are successively merged and transformed into new objects containing the desired data.

2.3.1 *ANR:* Analysis definition object

ID	*Project*
Type	*Status*

Analysis Name	*Description*
Model 1 Criteria	*Model 2 Criteria*
Model 1 Criteria ID	*Model 2 Criteria Id*
Model 1 Dir1 Name	*Model 2 Dir1 Name*
Model 1 Dir1 Type	*Model 2 Dir1 Type*
Model 1 Dir1 Person ID	*Model 2 Dir1 Person ID*
Model 1 Dir2 Name	*Model 2 Dir2 Name*
Model 1 Dir2 Type	*Model 2 Dir2 Type*
Model 1 Dir2 Person ID	*Model 2 Dir2 Person ID*
Model 1 From Person Name	*Model 2 From Person Name*
Model 1 From Person Type	*Model 2 From Person Type*
Model 1 From Person ID	*Model 2 From Person ID*
Model 1 From Organization Name	*Model 2 From Organization Name*
Model 1 From Organization Type	*Model 2 From Organization Type*
Model 1 From Org. Person ID	*Model 2 From Org. Person ID*
Model 1 To Person Name	*Model 2 To Person Name*
Model 1 To Person Type	*Model 2 To Person Type*
Model 1 To Person ID	*Model 2 To Person ID*
Model 1 To Organization Name	*Model 2 To Organization Name*
Model 1 To Organization Type	*Model 2 To Organization Type*
Model 1 To Org. Person ID	*Model 2 To Org. Person ID*
Model 1 Interaction	*Model 2 Interaction*
Final Criteria	*Final Criteria ID*
Final Elimination	*Final Interaction*
Final Connect	

2.3.2 *PRR:* Person record

The Person record contains a large number of data fields, such as *Age, Sex, Corporate Title, Telephone Number,* and up to 30 definable *Attributes.* For the purposes of the metrics calculations, only *ID* and *Name* are considered.

2.3.3 *ORR:* Organization record

The Organization record contains a large number of data fields, such as *Address,* various telephone numbers, and up to ten definable *Attributes.* For the purposes of the metrics calculations, only *ID* and *Name* are considered.

2.3.4 *DIE:* Data collection record object

Unique ID of record	*Project name*
Effective date	*Label*
Type (As-is, Should-be, etc.)	*Subtype (Mail, In person, Phone, etc.)*
From person ID	*From person name*
To person ID	*To person name*
Frequency of interaction	*Importance of interaction*
20 Impact values	*Notes*
Used?	

Used? is a Boolean value employed in the process of creating horizontal node points (see *2.6.3 Diagram 3: Creating horizontal node points,* page 36). It is initialized to "False."

2.3.5 *HNP:* Horizontal node point

"Unconfirmed":

Left person ID	*Right person ID (= Left person)*
Left person name	*Right person name (= Left person)*
Left person unit number (1 or 2)	*Right person unit number* (1 or 2)
Is left person confirmed? = True	*Is right person confirmed? = False*

"Confirmed":

Left person ID	*Right person ID*
Left person name	*Right person name*
Left person unit number (1 or 2)	*Right person unit number* (1 or 2)
Is left person confirmed? = True	*Is right person confirmed?* = True

The values of *Left person unit Number* and *Right person unit number* record whether the corresponding node is in unit 1 or unit 2, for use in the Integration metric (see *2.2.2.6 Integration,* page 11).

2.4 PROGRAM STRUCTURE

The metrics analysis performs the following basic steps:

1. Select the data collection records to be included in an analysis
2. Create *horizontal node points* representing the links between persons
3. Create a tree structure for the desired portion of the organization
4. Superimpose the links on the tree
5. Traverse the tree and calculate Clarity values for each node
6. Traverse the tree again and calculate values for the remaining metrics

2.4.1 Program flow

Diagram 1 (page 18) illustrates the top-level steps of the metrics analysis data flow. Subsequent diagrams will describe individual steps in greater detail, as indicated. For detailed discussion of the numbered items in the diagram, see *2.6 Detailed discussion,* page 33.

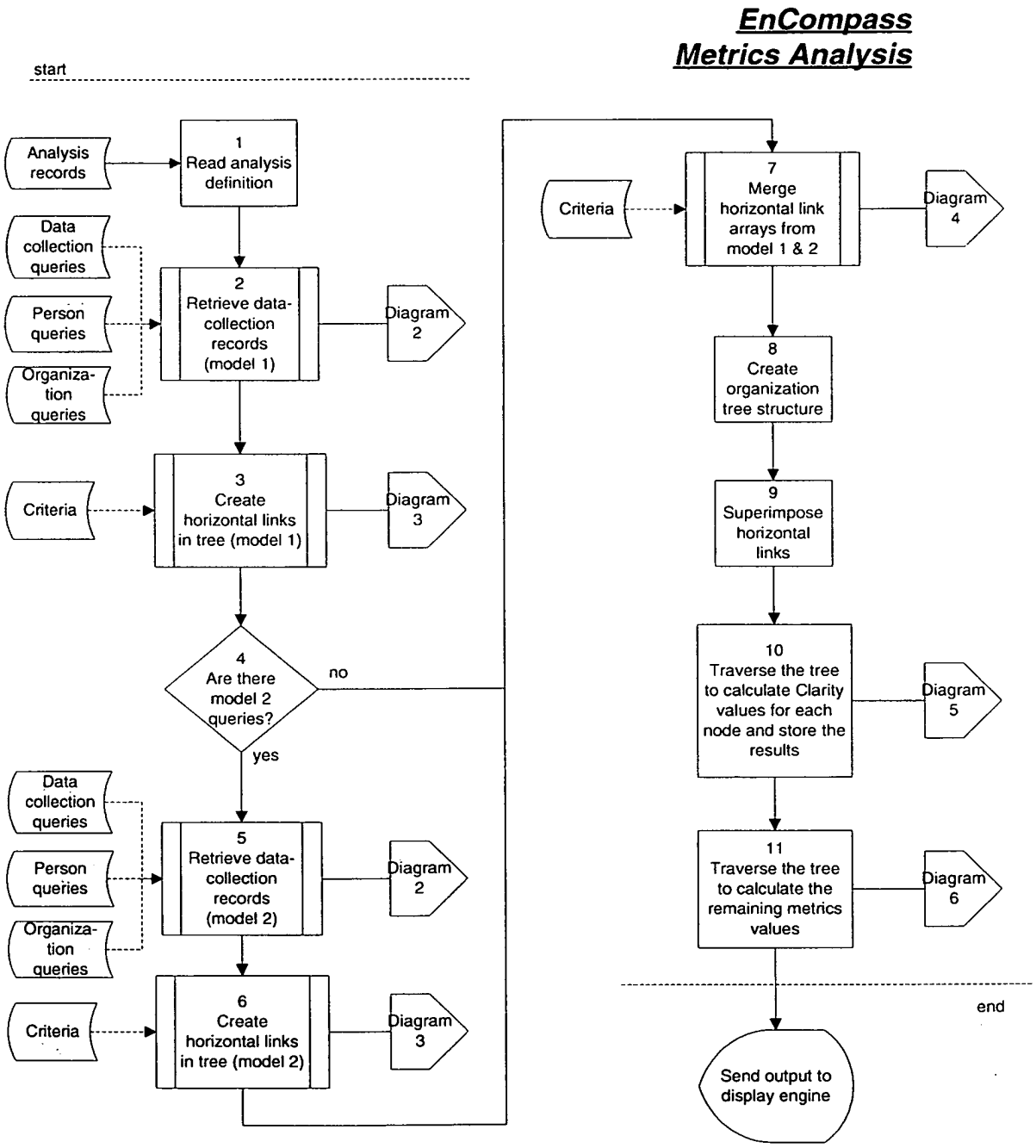

Diagram 1: EnCompass Metrics Analysis

2.4.2 Diagram 1: Top-level program structure

The following notes apply to the numbered items in diagram 1. For explanations of *key concepts,* see page 5.

1 Read the *analysis definition* from the database as an object *ANR* (see page 14). An analysis comprises one or more *database queries.* These must include at least one *data collection* query, which defines desired ranges of values for overall interaction frequency, overall importance, and impact figures on one or more specific issues. In addition, the definition may also query the following database columns:

From person *To person*

From organization *To organization*

The analysis must also include an *Agreement Status* setting and an associated *Criteria* query. These specify ranges of impact values for one or more issues, and determine whether the query will search for records in which the *From Person's* and the *To Person's* assessments either *Agree, Disagree,* or *Either* (agree or disagree) within the range.

In the EnCompass application interface, the user selects from lists of existing named queries, stored in the associated database tables (*Data Collection, Person, Organization,* and *Criteria*). The model 1 selection panel appears as in figure 6. (The panel for the optional model 2 part of the analysis definition is identical.)

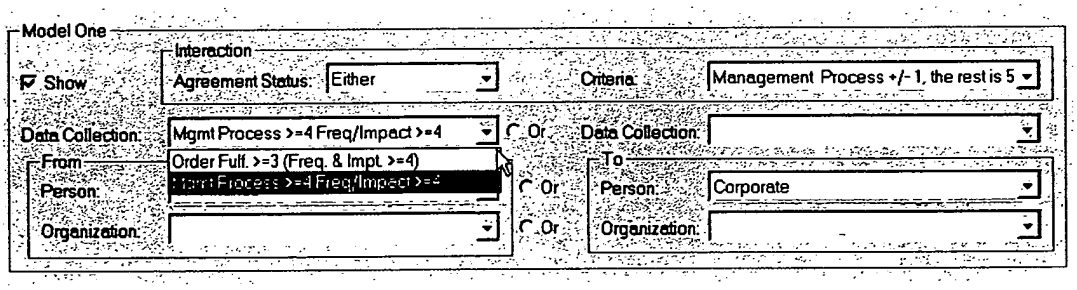

Figure 6: Model 1 query selection

For details, see *2.6.2 Diagram 2: Retrieving data collection records* (page 33), item 2.1.

2 Construct an array of *data collection records (DIE),* selecting records from the database which match the SQL queries in the model 1 analysis definition (diagram 2, page 23).

3 Match the array of records from step 2 against all other data collection records, transforming it into a new array of *horizontal node points* (class *HNP*). (Diagram 3, page 25.)

4 If there is a set of queries defined in the model 2 section of the analysis, process them the same as the model 1 queries. If there are no model 2 queries, proceed to 7.

5, 6 Duplicate steps 2 and 3 for the queries defined for model 2, if any. Create a separate *HNP* array for model 2.

7 Merge the *HNP* arrays from model 1 and model 2 into a single array, according to criteria specified in the analysis definition (see diagram 4, page 26). If there are no model 2 queries, this step simply transfers the model 1 *HNP*s. This step creates a set of horizontal links between nodes.

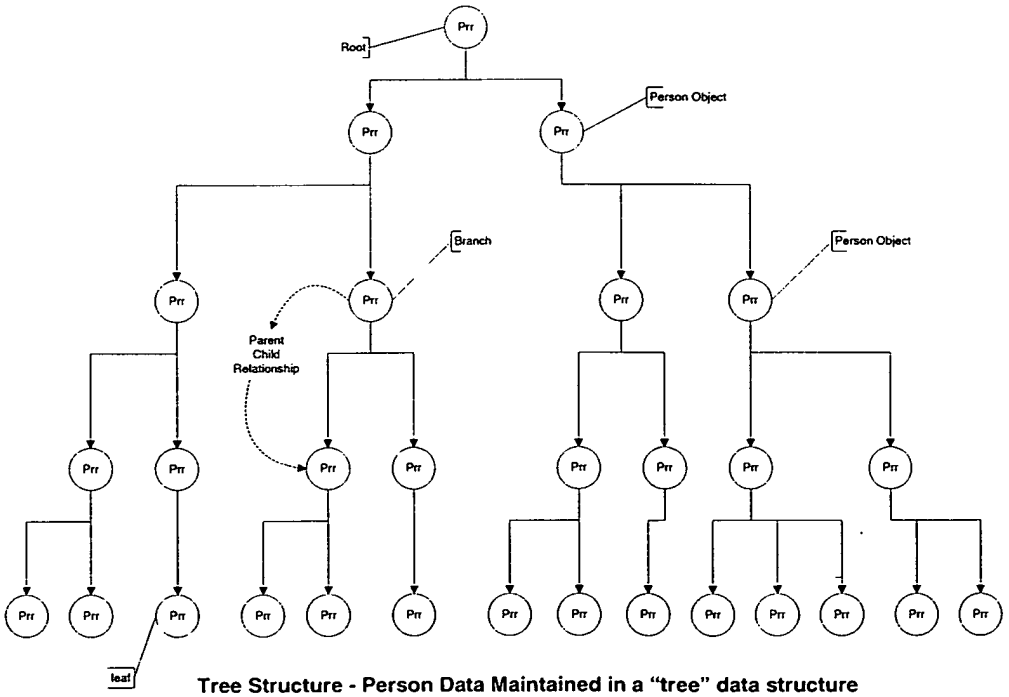

Figure 7: Basic tree structure

8 Create a tree structure for the organization, as shown in figure 7.

The tree is a hierarchical structure of person records (class *PRR*), arranged in a parent-child relationship, starting from a root node and including all its descendants.

9 Superimpose the horizontal node links from step 7 on the tree structure, as shown in figure 8.

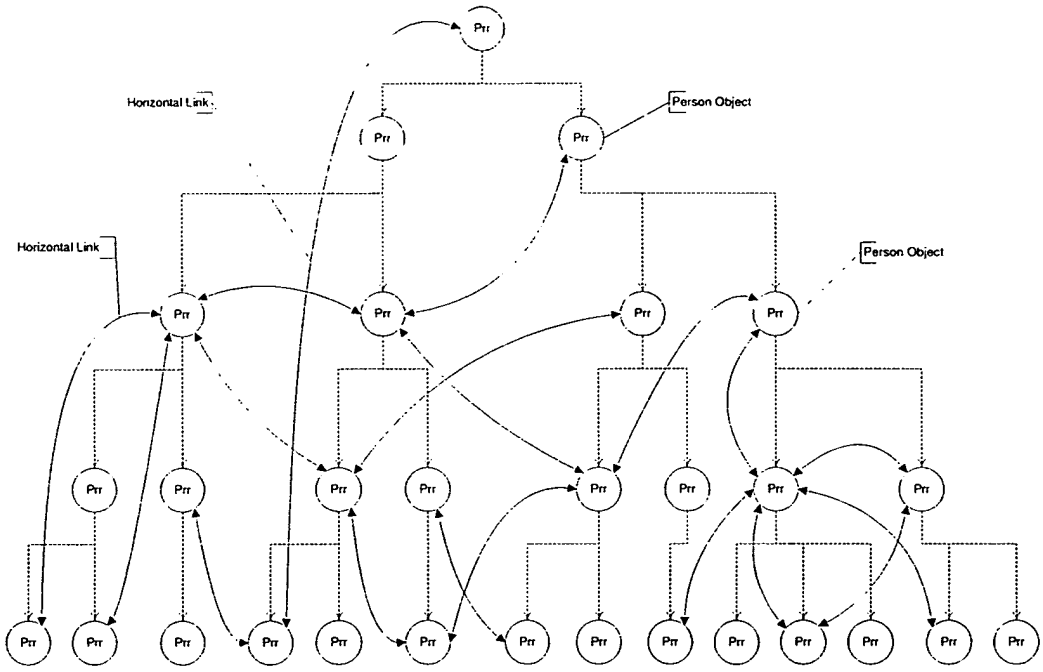

Figure 8: Superimposing horizontal links

10 Traverse the tree from the selected root node. To traverse a tree is to perform an operation once on every node in the tree, proceeding from one node to the next in one of a number of recognized patterns. Our traversal is in "IN" order, illustrated in figure 9 (page 22).

On the first traversal, count the number of confirmed and unconfirmed links and establish a Clarity value for each node. Store the results for use on the next traversal (see diagram 5, page 30).

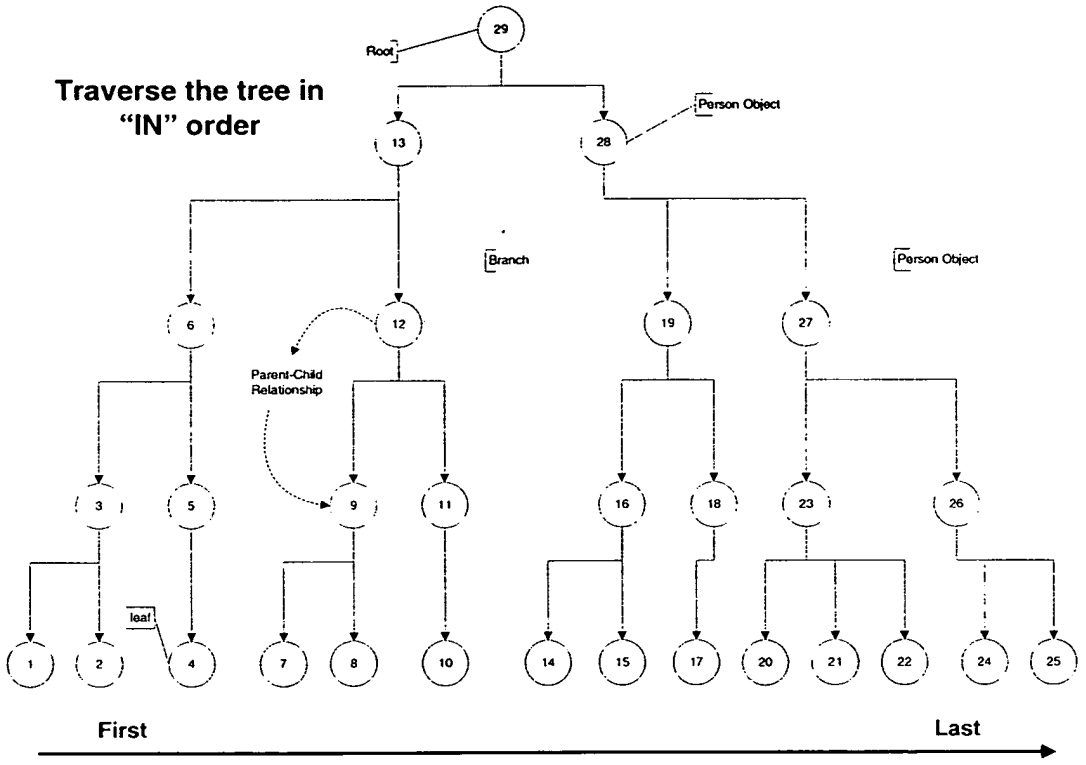

Figure 9: Traversing a tree in "IN" order

11 Traverse the tree again in "IN" order. On this second traversal, calculate the values of the remaining desired metrics (see diagram 6, page 31).

Display of results on the screen is performed by another system component (see *3.0 Display options,* page 52).

2.5 PROGRAM FLOW DIAGRAMS

The following diagrams illustrate the individual steps in the data collection process. For context, see diagram 1 (page 18). For discussion, see *2.6 Detailed discussion,* page 33.

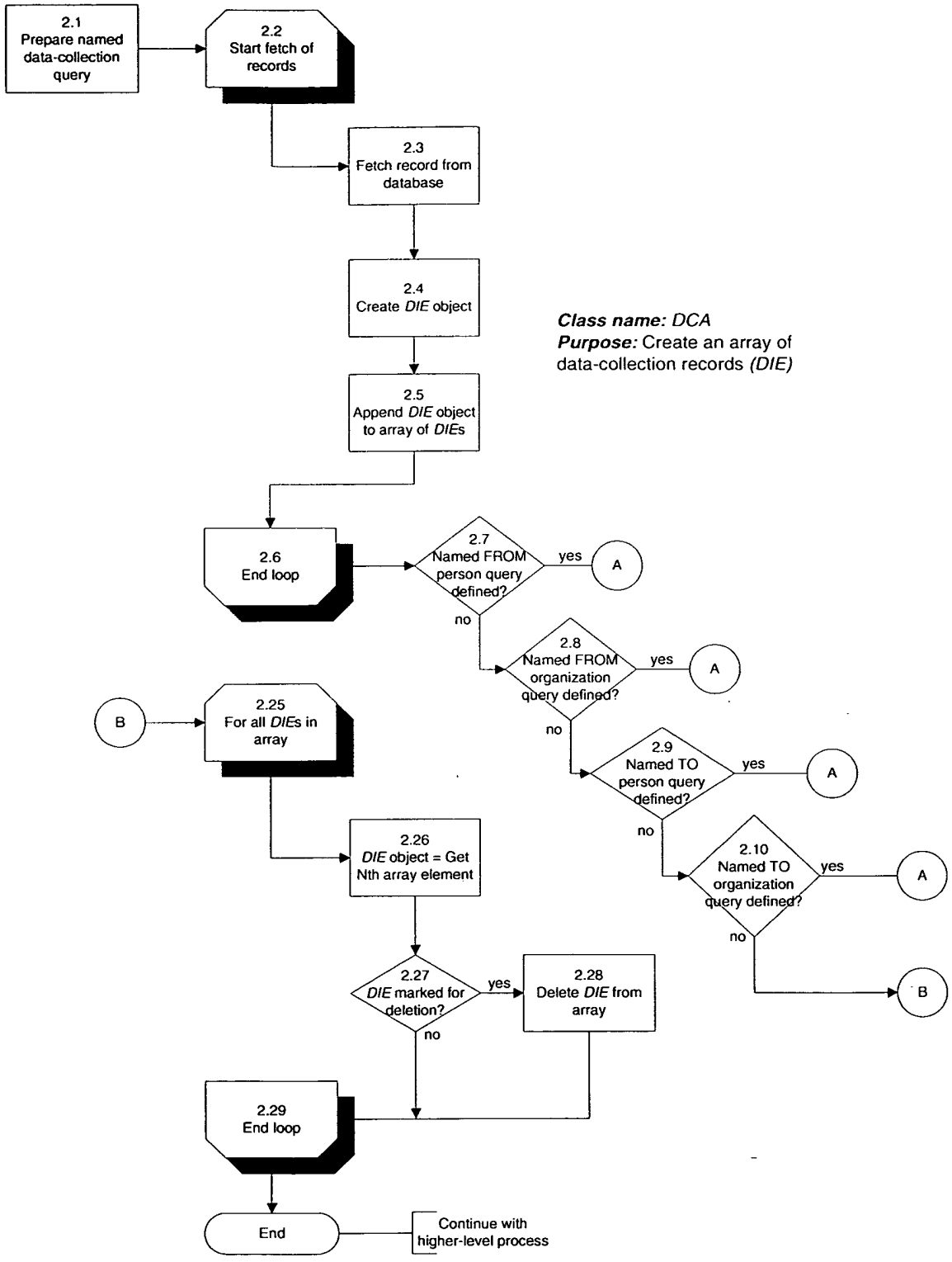

Diagram 2: Retrieving data collection records

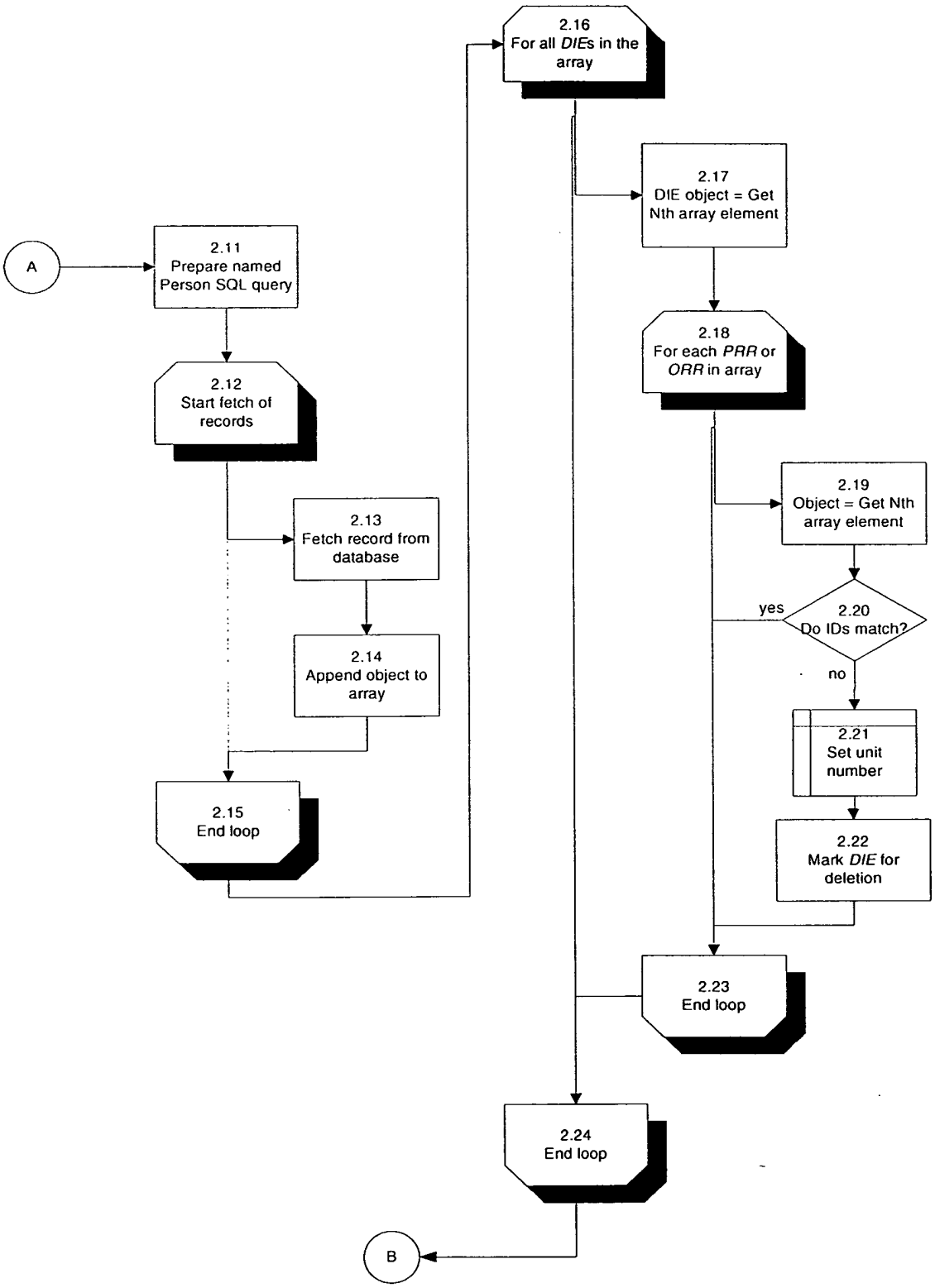

Diagram 2a: Creating data collection records, part 2

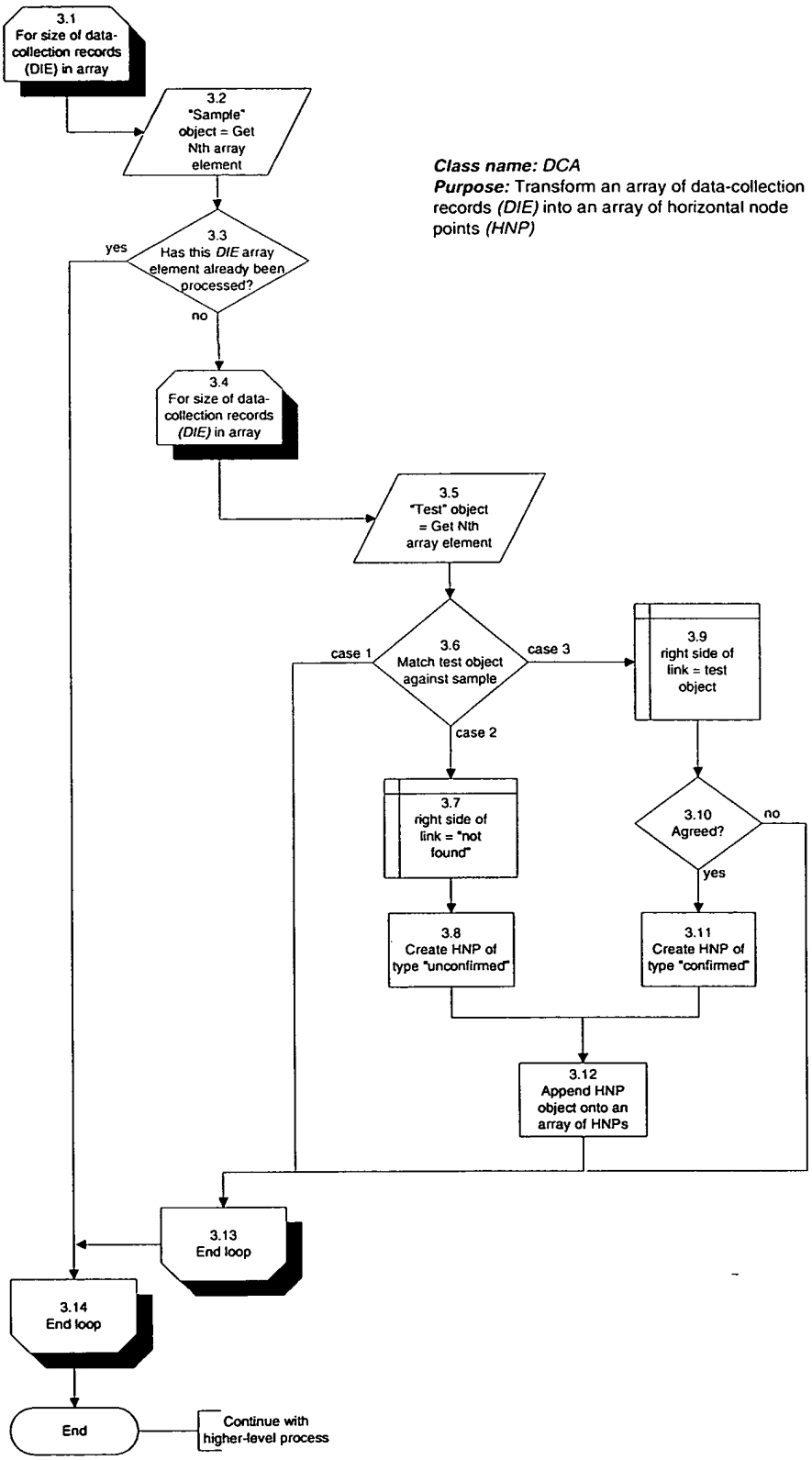

Diagram 3: Creating horizontal node points

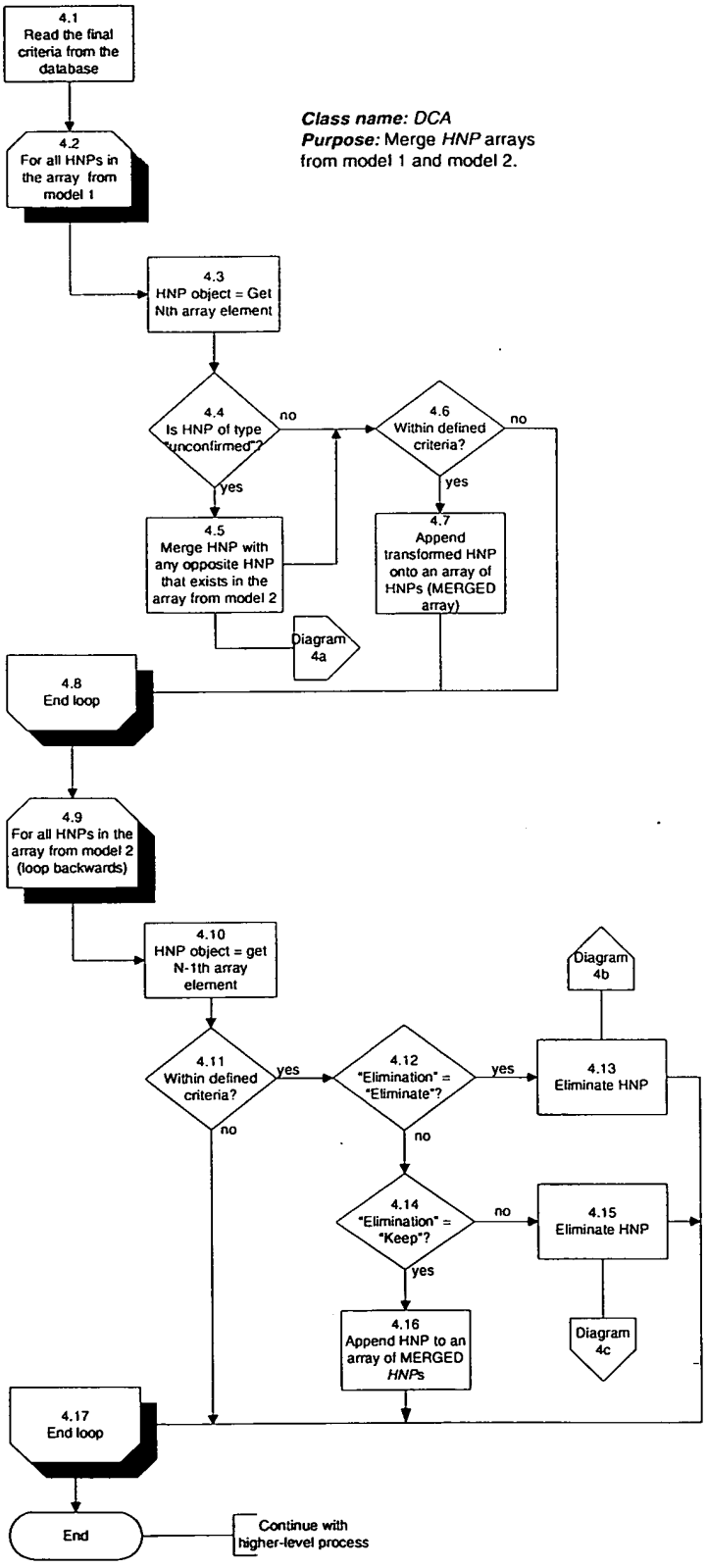

Diagram 4: Merging data collection models

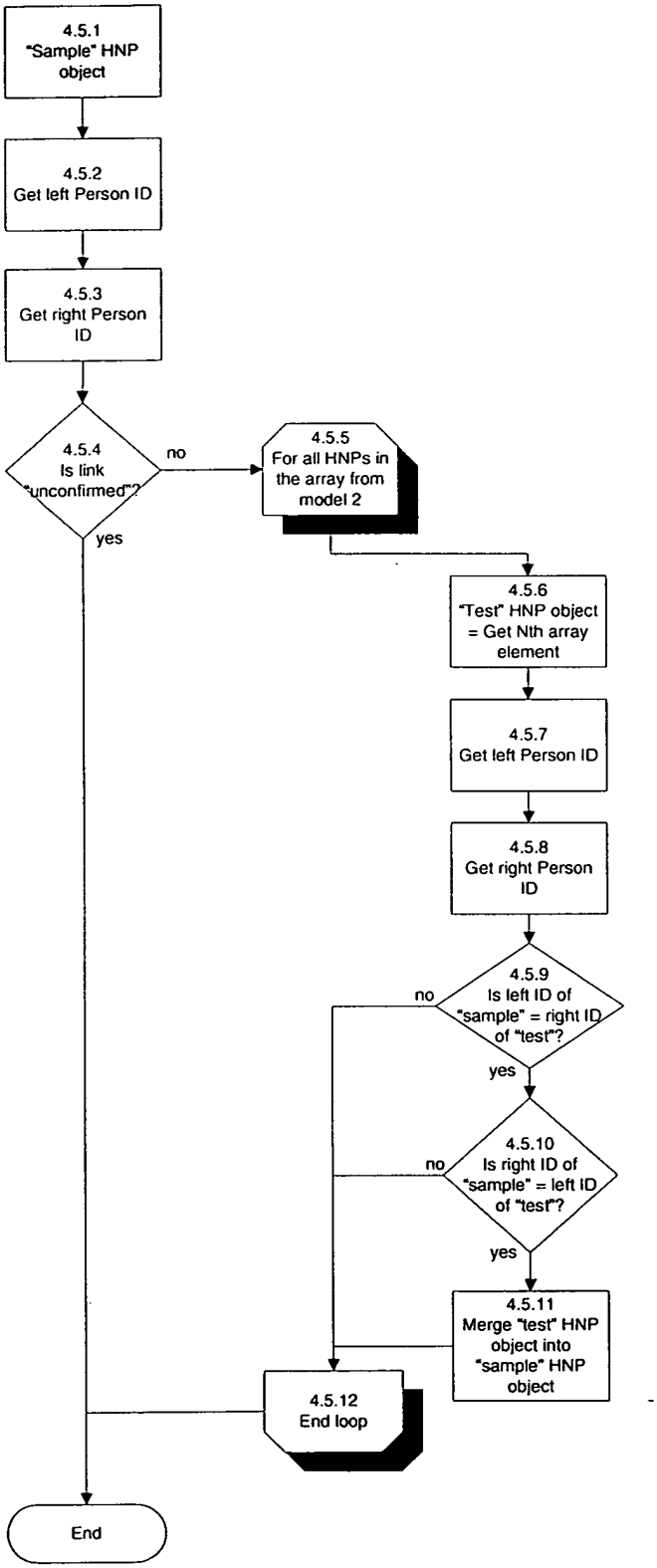

Diagram 4a: Merging model 2 HNPs

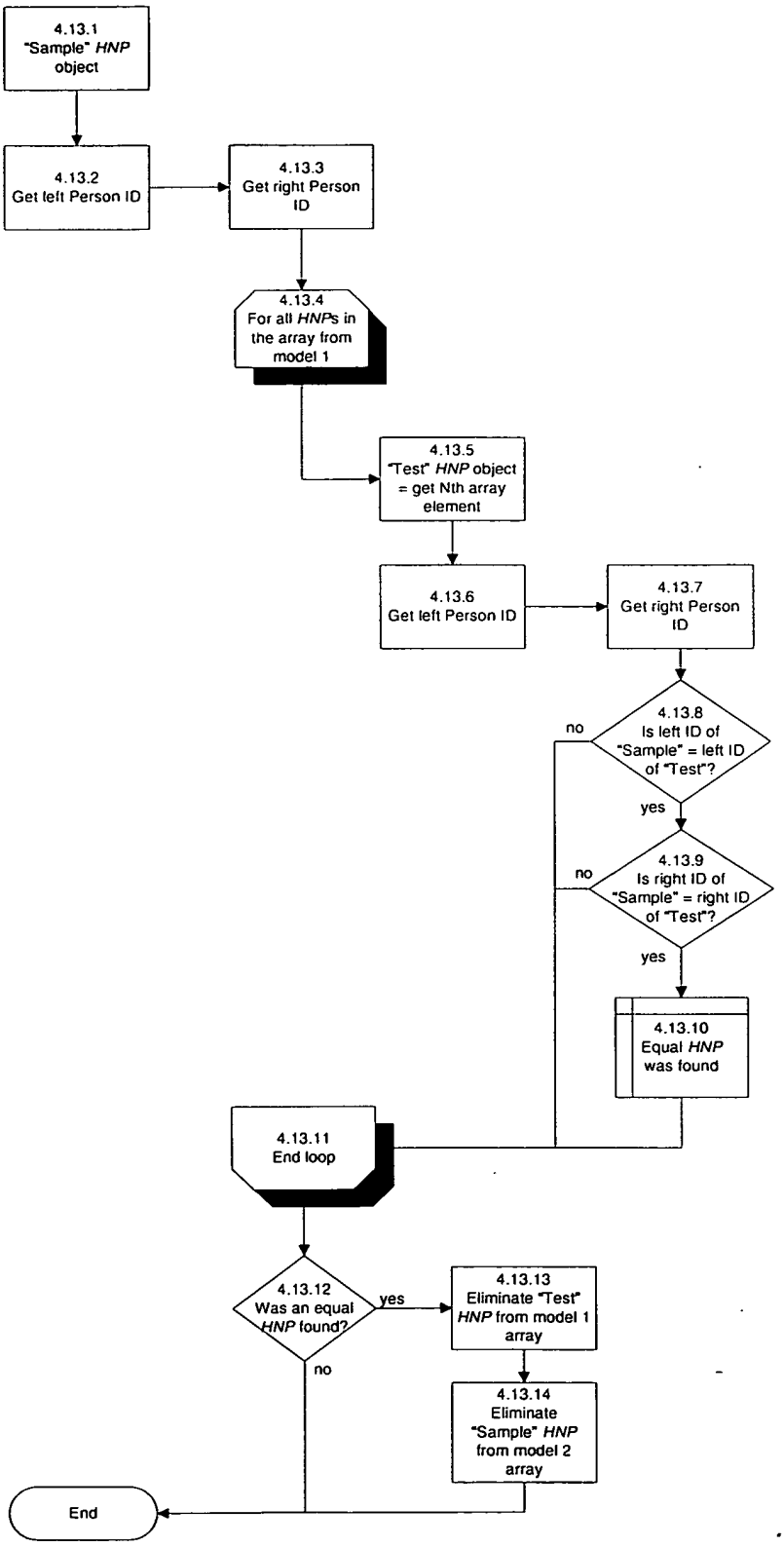

Diagram 4b: Eliminate if in both

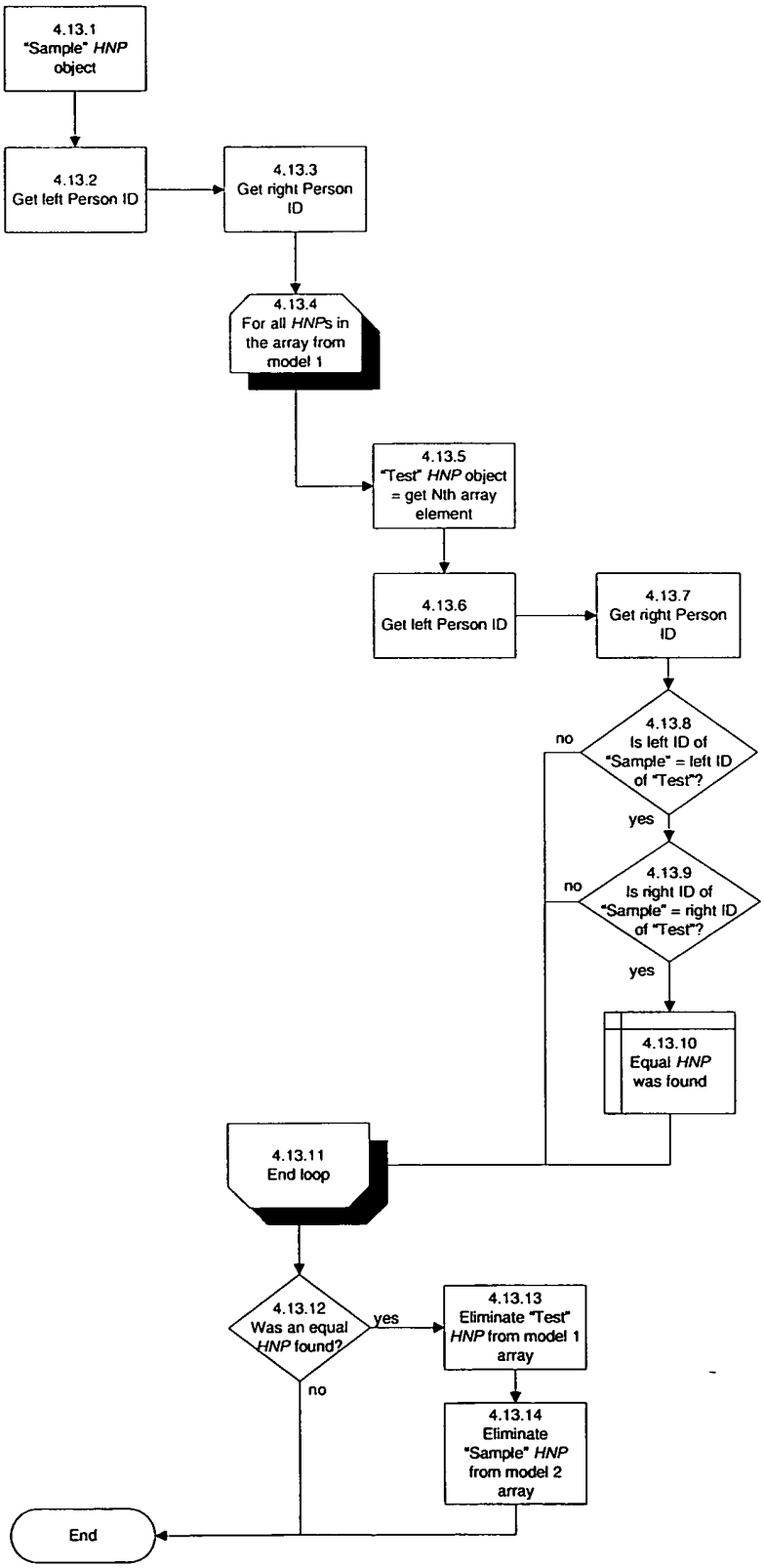

Diagram 4c: Keep if in both

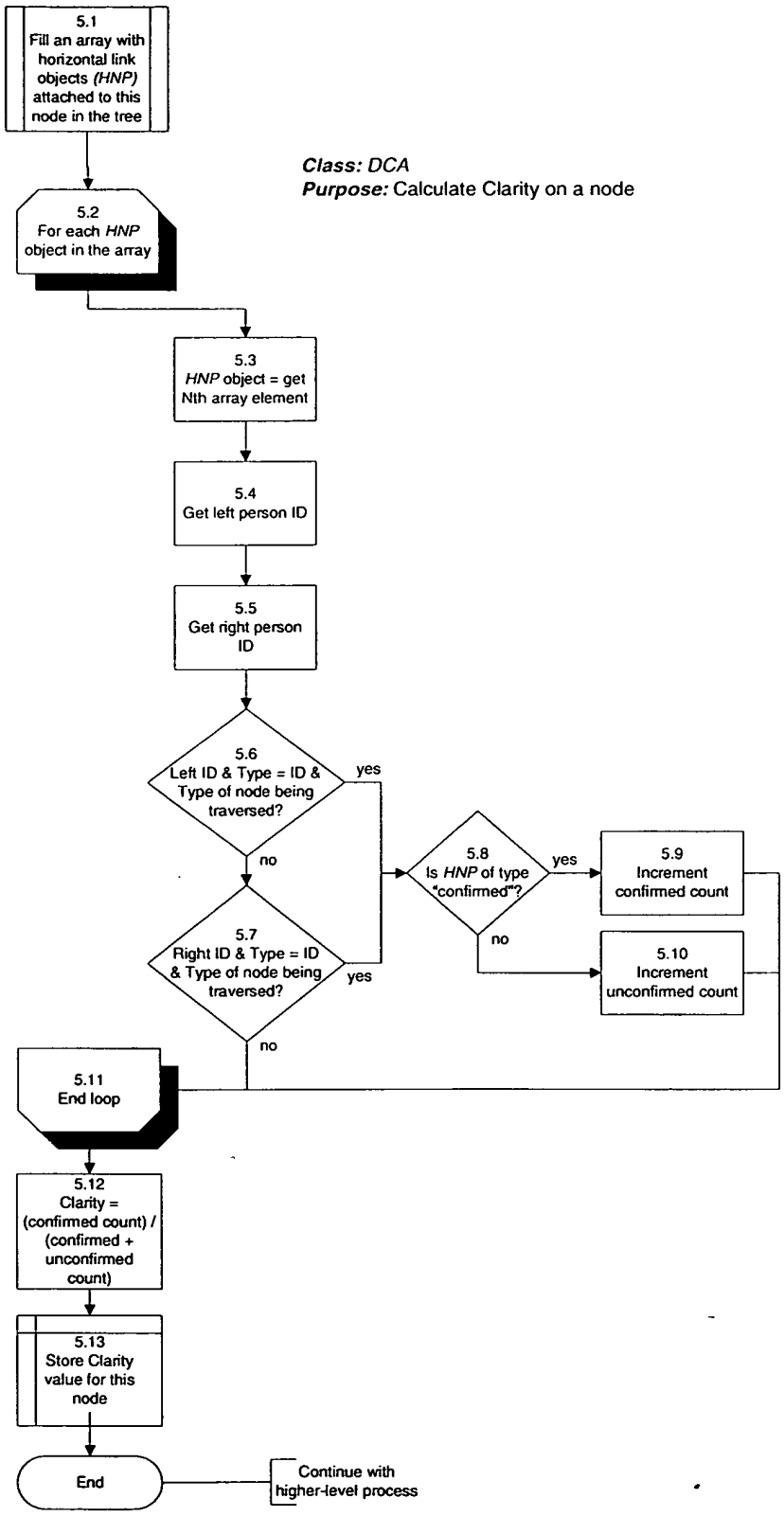

Diagram 5: Calculating node clarity

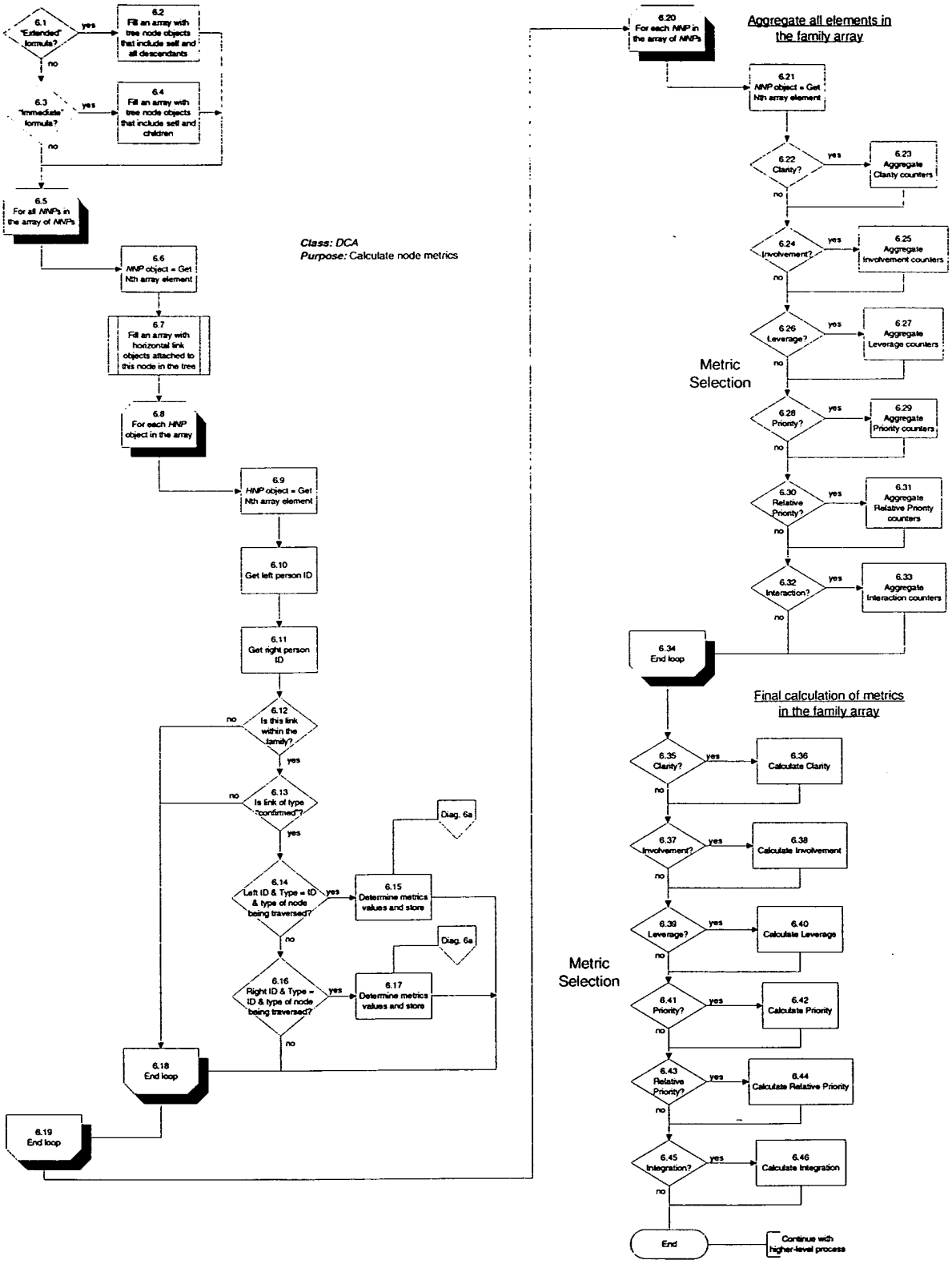

Diagram 6: Calculating metrics on a node

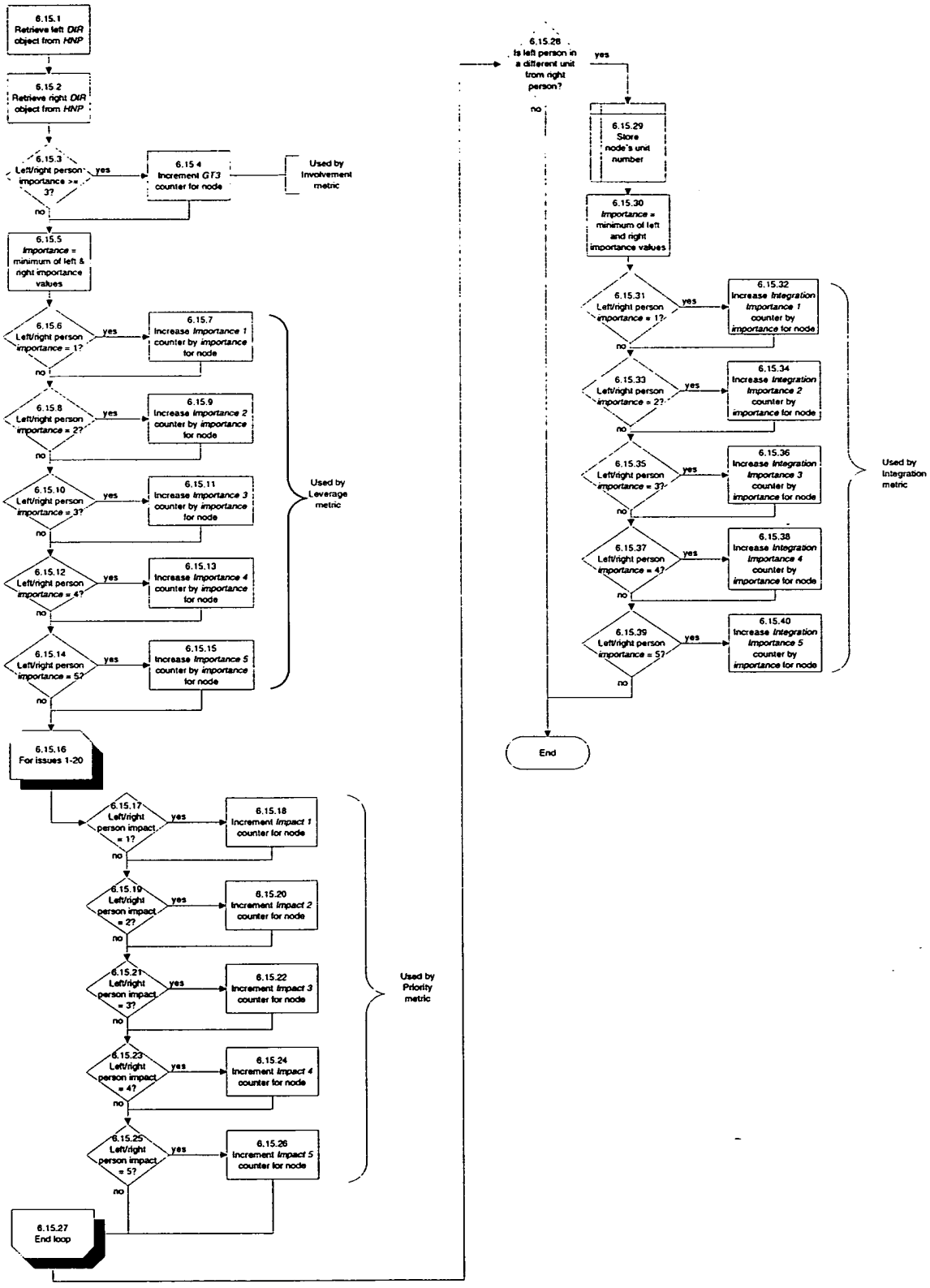

Diagram 6a: Calculating node parameters

2.6 DETAILED DISCUSSION

2.6.1 Introduction

Diagram 1 (page 18) and the associated discussion introduced the overall flow of the metrics analysis procedure. Diagrams 2-6 (pages 23-32) offered schematic views of the individual steps in the process. The following sections provide the annotations to diagrams 2-6. For definitions of *key concepts,* see page 5. For descriptions of data objects (e.g. *DIE*), see page 13.

2.6.2 Diagram 2: Retrieving data collection records

After reading the *analysis definition* to determine what *SQL queries* will be called for (diagram 1, item 1), the first step in peforming the data analysis is to retrieve the matching records from the database (see diagram 2, page 23).

2.1 Parse the individual query conditions from the analysis definition and construct a SQL statement to retrieve the matching data collection records.

For example, an analysis might use a data collection query such as: "Frequency and Importance ≥ 4; Management Processes ≥ 4," thus searching for interactions with a high or very high overall frequency and importance, and a high or very high impact on the defined issue of "management processes" (internally, issue #6). If the current project is called "Sample," this query will generate the following SQL:

SELECT * FROM Data_Collection WHERE (ID LIKE '%') AND (PROJECT = 'Sample') AND (PERSONFREQUENCY >= 4) AND (PERSONIMPORTANCE >= 4) AND (IMPACTVAL6 >= 4) ORDER BY ID

2.2. *Loop:* Perform steps 2.3-2.6 for every data collection record found.

2.3-2.4 Retrieve the next record and store it in memory as a data collection object (class *DIE;* see page 15).

2.5 Append the *DIE* object to an array of *DIE*s.

2.6 Go to the next record.

2.7 Test whether the analysis definition includes an optional *From Person* query. If "true," go to step 2.11. If there is none, go to the next test (2.8).

2.8 Test whether the analysis definition includes an optional *From Organization* query. If "true," go to step 2.11. If there is none, go to the next test (2.9).

2.9 Test whether the analysis definition includes an optional *To Person* query. If "true," go to step 2.11. If there is none, go to the next test (2.10).

2.10 Test whether the analysis definition includes an optional *From Organization* query. If "true," go to step 2.11. If there is none, go to 2.24.

[Steps 2.11 through 2.23 form a procedure that may be repeated up to four times, once for each of the optional queries included in the analysis definition. The values used each time will depend on the type of query, as noted. (Refer to diagram 2a [page 24].)]

2.11 From the conditions in the analysis definition (model 1 or model 2), construct a SQL database query of the appropriate type. For example, an analysis might include a *From Person* query named "Executive," which searches for cases in which the person reporting the interaction is in the executive division. Again, if the project name is "Sample," this query generates the following SQL:

WHERE (ID LIKE '%') AND (PROJECT = 'Sample') AND (DIVISION = 'Executive') ORDER BY ID

From Person and *To Person* queries will retrieve records from the *person* table; *From Organization* and *To Organization* queries will search the *organization* table.

2.12 *Loop:* Perform steps 2.13-2.15 for every record in the appropriate table (*person* or *organization*).

2.13 Retrieve the next record as an object of type *PRR* (for person queries) or *ORR* (for organization queries).

2.14 Append the *PRR* or *ORR* to an array of *PRR*s or *ORR*s.

[Steps 2.16 through 2.23 compare the objects in the array of person records *(PRR)* or organization records *(ORR)* just created against those in the *DIE* array, searching for matching record ID values. Nonmatching records are flagged for later deletion.]

2.16 *Loop 1:* Test every element in the *DIE* array as follows:

2.17 Get the next element in the array.

2.18 *Loop 2:* For every element in the (*PRR* or *ORR*) array from 2.12, do the following:

2.19 Get the next array element.

2.20 Test the *ID* values in the two objects according to the following scheme:

for query type:	**test *DIE* value:**	**against ID in:**
"From Person"	*From Person ID*	*PRR*
"From Organization"	*From Organization ID*	*ORR*
"To Person"	*To Person ID*	*PRR*
"To Organization"	*To Organization ID*	*ORR*

If the values match, retain the *DIE* object and go to the next element (2.23). If there is no match, go to 2.21.

2.21 Record which organizational unit the node resides in, for use in the Integration metric (*2.2.2.6 Integration,* page 11). In model 1, if the node matches the "From Person," it is in unit 1. If the the node matches the "To Person," it is in unit 2. In model 2, the unit assignments are reversed.

2.22 Mark the *DIE* object for deletion.

2.23-2.24 After testing all *PRR* or *ORR* records against every *DIE* for the current query type, check for the presence of the next type (2.7-2.10). When all types have been tested, go to 2.25.

(For steps 2.25 through 2.29, refer back to diagram 2.)

2.25 *Loop:* Perform the remaining steps for every element in the *DIE* array.

2.26 Get the next array element.

2.27 Check whether the object was marked for deletion in 2.21.

2.28-2.29 If it is marked, delete it from the array (2.28). If not, retain it and go to the next *DIE*. When all *DIE*s have been checked, return to the main procedure (diagram 1, item 3 or 6).

2.6.3 Diagram 3: Creating horizontal node points

After selecting *data collection records* from the database, according to queries specified in the *analysis definition,* the program transforms the array of data collection records into **horizontal node points** (class *HNP*). (See diagram 3, page 25.) A *HNP* object links an interaction reported by one person (the *left person*) with one reported by a second person (the *right person*). The *From person* and *To person* names on both sides of the *HNP* link are compared end-for-end: if they match in both directions, the link is "confirmed", meaning that the right person also says that the interaction takes place (though the frequency, importance, and impact values may differ). If only in one direction, it is an "unconfirmed" interaction.

3.1 Begin with an array of *data collection records* (DIE) (diagram 1, item 2).

Loop 1: Perform the following steps on every element in the *DIE* array.

3.2 Get the next element in the array. Call this object "Sample." (Values for the left side of the *HNP* link will come from this object.)

3.3 Check the value of *Used?* If "true," this item has already been processed. Proceed to the next array element.

3.4 *Loop 2:* For each iteration of loop 1, perform the following steps on every element in the *DIE* array.

3.5 Get the next item in the array to compare against "Sample." Call this object "Test."

3.6 (a) Match the *From person ID* in "Sample" against the *To person ID* in "Test."

Case 1: If the match fails, discard "Test" (go to 3.13).

If the match is true, (b) match the *From person ID* in "Test" against the *To person ID* in "Sample."

Case 2: If match (b) fails, the link is *unconfirmed.* Go to 3.7.

Case 3: If match (b) is true, the link is *confirmed.* Go to 3.9.

3.7 Store the fact that there is no match for the right side of the link.

3.8 Create an "unconfirmed" horizontal node point object *(HNP)* (see page 15). Populate the *Right person ID* and *Right person name* with dummy values, copied

from "Sample." The *Right person confirmed* value is unused here and will be discarded later. Go to 3.12.

3.9 Store the values from the test object. This will be the right side of the horizontal node point *(HNP)*.

3.10 Compare the issue-impact values in the sample and test objects. If they fall within the agreement criteria in the analysis definition, go to 3.11. If not, exit loop 2 (go to 3.13).

3.11 Create a "confirmed" horizontal node point object *(HNP)* (see page 15). The *Right person confirmed value* is unused here and will be discarded later.

3.12 Append the new *HNP* object ("confirmed" or "unconfirmed") to the array being built.

3.13 Go to the next test object.

3.14 Go to the next sample object. When all data collection records in the *DIE* array have been examined, return to the main procedure (diagram 1, item 4).

2.6.4 Diagram 4: Merging data collection models

Every analysis must include at least one set of queries (model 1) to retrieve data collection records from the database; the program stores the results in an array of horizontal node points *(HNP;* see page 15*)*. Optionally, an analysis may also include a second set of queries for *model 2*, the results of which are stored in a second *HNP* array. In this case, the results of the two query sets must be merged into a single *HNP* array for further processing (diagram 4, page 26).

4.1 The parameters for merging the model 1 and model 2 arrays are set by the user, in the *Show Results* panel in the program interface (figure 10). The settings are stored in the database for reuse.

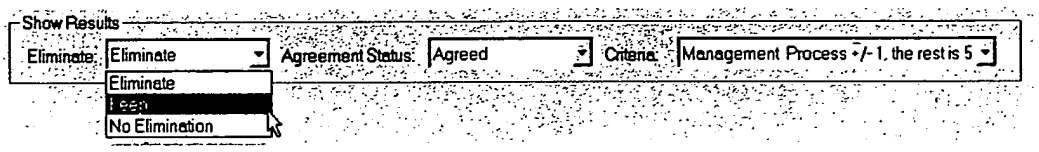

Figure 10: *Show Results* panel

The *Eliminate* setting determines what happens to records that appear in both models, by specifying the action to take when two compared *HNP* objects contain the same pair of left and right person IDs (in either order and regardless of impact values). Options are *eliminate, keep,* and *no elimination.*

- "Eliminate" discards both records (binary subtraction).
- "Keep" retains both records (binary addition).
- "No elimination" performs no checks and retains any record that appears in either model.

Agreement Status may be *agreed, disagreed,* or *either.*

- "Agreed" retains the records if their impact values match within the range specified by the Criteria setting.
- "Disagreed" retains the records if their impact values do NOT match within the range specified by the Criteria setting.
- "Either" performs no checking of issue-impact values.

Criteria specify ranges of issue-impact values for the *agreement status* condition.

4.2 *Loop:* Perform steps 4.3-4.7 for every element in the array of *HNP*s from model 1. This loop will move the model 1 *HNP*s into the new merged *HNP* array.

4.3 Get the next array element as a new *HNP* object.

4.4 The object was created as either type "confirmed" or "unconfirmed" (diagram 3, steps 3.8, 3.11). If it is "unconfirmed," go to 4.5. If not, go to 4.6.

4.5 Check the *HNP* object against the array from model 2 to see whether a match for the right side of this link exists there. If it does, merge the *HNP*s to transform this into a "confirmed" link (see *2.6.4.1 Diagram 4a: Merging model 2 HNPs,* page 39).

4.6 Check the impact values in the *HNP* object against the ranges specified in the "Criteria" query and the "Agreement Status" condition (4.1). If they do not match, go to the next object (4.8). If they match, go to 4.7.

4.7 Append the *HNP* (now of type "confirmed") to a new array of *merged HNP*s.

4.8 Get the next *HNP* object from the model 1 array. When all model 1 *HNP*s have been examined, continue to 4.9.

4.9 *Loop:* Perform the remaining steps for every element in the array of *HNP*s from model 2. (Loop backward from the end of the array, in order to avoid reindexing when elements are deleted.) This loop will merge the model 2 *HNP*s into the new merged array.

4.10 Get the next element in the array as an object of type *HNP.*

4.11 Check the impact values in the *HNP* object against the ranges specified in the "Criteria" query and the "Agreement Status" condition (4.1). If they do not match, get the next object (4.17). If they match, go to 4.12.

4.12 Check the "Eliminate" parameter (4.1). If it is "Eliminate," go to 4.13. If not, go to 4.14.

4.13 See *2.6.4.2 Diagram 4b: Eliminate if in both,* page 40.

4.14 Check the "Eliminate" parameter (4.1). If it is "Keep," go to 4.15. If not, go to 4.16.

4.15 See *2.6.4.3 Diagram 4c: Keep if in both,* page 41.

4.16 The *HNP* object from model 2 meets all the criteria for merging. Add it to the *merged HNP* array.

4.17 Get the next element in the model 2 array. When finished, return to the main procedure (diagram 1, item 8).

2.6.4.1 Diagram 4a: Merging model 2 HNPs

The type "unconfirmed" indicates that the *From Person* and *To Person* values on the left side of a horizontal node point *(HNP)* do not match those on the right side. It is always possible, however, that a match will exist in a data collection record returned by a model 2 query. In this case, the link must be added to the array of merged *HNP*s. (Refer to diagram 4a, page 27.)

4.5.1 Call the *HNP* object "Sample."

4.5.2 Read the value of *Left person ID* from "Sample."

4.5.3 Read the value of *Right person ID* from "Sample."

4.5.4 The object was created as either type "confirmed" or "unconfirmed" (diagram 3, steps 3.8, 3.11). If it is "unconfirmed," return to the main procedure (4.6). If not, go to 4.5.5.

4.5.5 *Loop:* Test "Sample" against every *HNP* in the array from model 2.

4.5.6 Get the next object in the array. Call it "Test."

4.5.7 Read the value of *Left person ID* from "Test."

4.5.8 Read the value of *Right person ID* from "Test."

4.5.9 Match the *Left person ID* in "Sample" against the *Right Person ID* in "Test." If the match is false, go to the next Test object (4.5.12). If true, go to 4.5.10.

4.5.10 Match the *Right person ID* in "Sample" against the *Left Person ID* in "Test." If the match is false, go to the next Test object (4.5.12). If true, go to 4.5.11.

4.5.11 Combine "Sample" and "Test" into a single "confirmed" *HNP* object.

4.5.12 Get the next "Test" object from the model 2 array. When all *HNP*s have been tested, return to the main procedure (4.6).

2.6.4.2 Diagram 4b: Eliminate if in both

When the *Eliminate* setting is "Eliminate" (4.1), any *HNP* object found in both the model 1 and model 2 arrays is discarded. (Refer to diagram 4b, page 28.)

4.13.1 Get the *HNP* object from the model 2 array. Call it "Sample."

4.13.2 Read the value of *Left Person ID* from "Sample."

4.13.3 Read the value of *Right Person ID* from "Sample."

4.13.4 *Loop:* Perform steps 4.13.5-4.13.10 for every element in the array of *HNP*s from model 1.

4.13.5 Get the next *HNP* object in the array. Call it "Test."

4.13.6 Read the value of *Left Person ID* from "Test."

4.13.7 Read the value of *Right Person ID* from "Test."

4.13.8 Compare the *Left Person ID* in "Sample" against the *Left Person ID* in "Test." If they match, go to 4.13.9. If not, go to the next Test object.

4.13.9 Compare the *Right Person ID* in "Sample" against the *Right Person ID* in "Test." If they match, go to 4.13.10. If not, go to the next Test object.

4.13.10 Record the fact that the model 1 and model 2 *HNP*s match.

4.13.11 Get the next *HNP* in the model 1 array. When all *HNP*s have been tested, go to 4.13.12

4.13.12 Check the result of step 4.13.10 to see whether an equal *HNP* was found. If no, exit. If yes, go to 4.13.13.

4.13.13 Delete the *HNP* object "Test" from the model 1 array.

4.13.14 Delete the *HNP* object "Sample" from the model 2 array. Return to 4.17.

2.6.4.3 Diagram 4c: Keep if in both

When the *Eliminate* setting is "Keep" (4.1), any *HNP* object found in both the model 1 and model 2 arrays is retained. (Refer to diagram 4c, page 29.)

4.15.1 Get the *HNP* object from the model 2 array. Call it "Sample."

4.15.2 Read the value of *Left Person ID* from "Sample."

4.15.3 Read the value of *Right Person ID* from "Sample."

4.15.4 *Loop:* Perform steps 4.15.5-4.15.10 for every element in the array of *HNP*s from model 1.

4.15.5 Get the next *HNP* object in the array. Call it "Test."

4.15.6 Read the value of *Left Person ID* from "Test."

4.15.7 Read the value of *Right Person ID* from "Test."

4.15.8 Compare the *Left Person ID* in "Sample" against the *Left Person ID* in "Test." If they match, go to 4.15.9. If not, go to the next Test object.

4.15.9 Compare the Right Person ID in "Sample" against the Right Person ID in "Test." If they match, go to 4.15.10. If not, go to the next Test object.

4.15.10 Record the fact that the model 1 and model 2 *HNP*s match.

4.15.11 Get the next *HNP* in the model 1 array. When all *HNP*s have been tested, go to 4.15.12

4.15.12 Check the result of step 4.15.10 to see whether an equal *HNP* was found. If yes, exit. If no, go to 4.15.13.

4.15.13 Delete the *HNP* object "Test" from the model 1 array.

4.15.14 Delete the *HNP* object "Sample" from the model 2 array. Return to 4.17.

2.6.5 Diagram 5: Calculating node clarity

Perform the following steps for each node in the tree. First, count the number of "confirmed" and "unconfirmed" links.

5.1 List the horizontal node points (*HNP*s) that are attached to this node and place them in an array. (An existing named function carries out this operation.)

5.2 *Loop:* Perform steps 5.3-5.10 for each *HNP* element in the array.

5.3 Get the next element in the array.

5.4 Read the value of *Left Person ID.*

5.5 Read the value of *Right Person ID.*

5.6 Compare the values of *Left Person ID* and *Table #* in the *HNP* object being examined against the ID and table number of the node being traversed. If they match, go to 5.8. If not, go to 5.7.

5.7 Compare the values of *Right Person ID* and *Table #* in the *HNP* object being examined against the ID and table number of the node being traversed. If they match, go to 5.8. If not, go to the next array element (5.11).

5.8-5.10 The *HNP* object was created as either type "confirmed" or "unconfirmed" (see diagram 3, items 3.8 and 3.11). If it is "confirmed," increment the count of "confirmed" links. If not, increment the count of "unconfirmed" links.

5.11 Get the next element in the array. When all array elements have been examined, exit to step 5.12.

Next, calculate and store the Clarity value for the node:

5.12 Calculate the clarity value for this node, using the formula:

$$Clarity = \frac{Links(\text{confirmed})}{Links(\text{confirmed}) + Links(\text{unconfirmed})}$$

5.13 Store the resulting figure for later use. Return to the main procedure (diagram 1, step 12).

2.6.6 Diagram 6: Calculating metrics values

The following procedure calculates the remaining metrics values for the node being traversed.

The first step is to list the relevant population of tree node points. For most metrics, the user can elect to take the measurement with respect to only those who report to him or her directly (**immediate** metrics), or against the collection of all subordinates (**extended** metrics). In the first case, the procedure counts only the node's direct children. In the second case, the measurement includes all descendants to the end of the organization structure.

[Steps 6.2 and 6.4 create arrays of *tree node points* (class *NNP*). These objects are abstract representations of nodes in a three-dimensional space, which inherit the *ID* value of the record connected with the node point. In most cases this is a person record *(PRR),* although it may be any other type of database record that can be attached to a tree structure.]

6.1 If the desired metric is an "extended" metric, go to 6.2. If not, go to 6.3.

6.2 Create an array of tree node points including the node being traversed and all its descendants.

6.3 If the measurement is an "immediate" metric, go to 6.4.

6.4 Create an array of tree node points including the node being traversed and its child nodes.

6.5 *Loop 1:* For every element in the array of node points, perform the following steps.

6.6 Get the next element in the array.

6.7 List the horizontal node points (*HNP*s) that are attached to this node and place them in an array. (An existing named function carries out this operation.)

6.8-6.9 *Loop 2:* Perform the following steps for every element in the array from 6.7.

6.10 Retrieve the value of *Left Person ID.*

6.11 Retrieve the value of *Right Person ID.*

6.12 Determine whether the person on the other end of the link is within the "family." For "extended" metrics, the family includes the node being analyzed and all of its descendant nodes (fig. 11). For "immediate" metrics, it includes only the node being analyzed and its direct child nodes (fig. 12). Links that are not within the family are discarded.

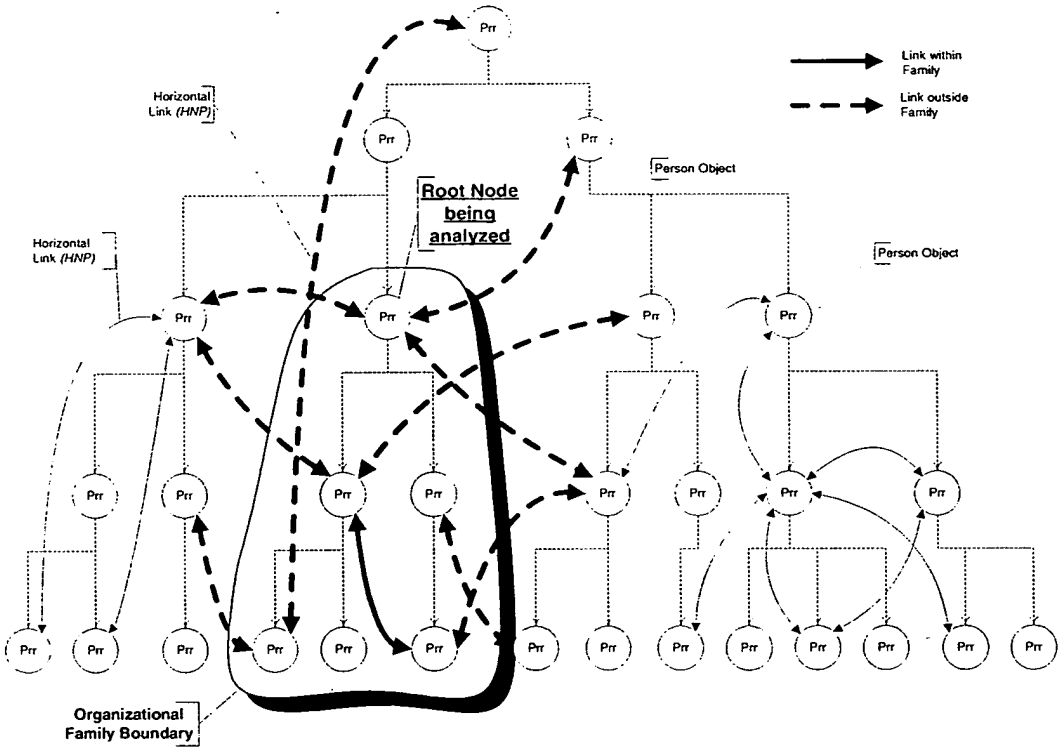

Figure 11: Extended node family

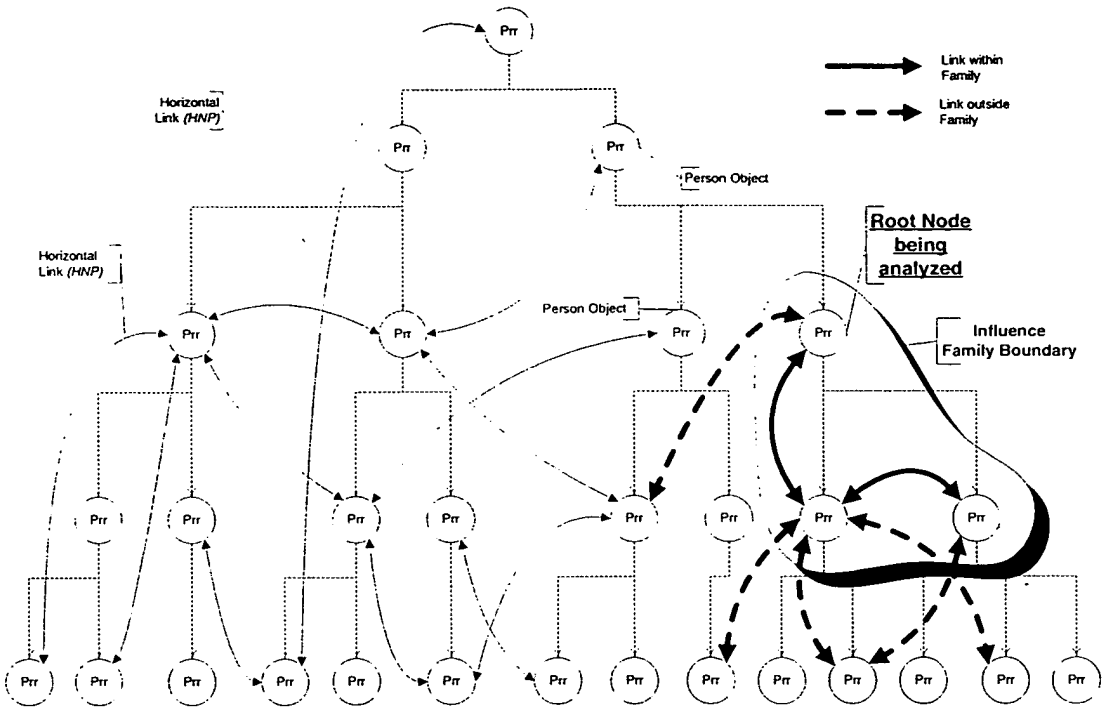

Figure 12: Immediate node family

6.13 The link was created as either type "confirmed" or "unconfirmed" (see diagram 3, items 3.8 and 3.11). If it is not "confirmed," discard and go to the next element in the array (6.18).

6.14 Compare the values of *Left Person ID* and *Table #* in the *HNP* object being examined against the ID and table number of the node being traversed. If they match, go to 6.15. If not, go to 6.16.

6.15 Determine and store the variables for the requested metrics (see diagram 6a, page 32; discussion at *2.6.6.1 Diagram 6a: Calculating metrics parameters,* page 49).

6.16 Compare the values of *Right Person ID* and *Table #* in the *HNP* object being examined against the ID and table number of the node being traversed. If they match, go to 6.17. If not, go to the next array element (6.18).

6.17 Determine and store the values of the requested metrics for the selected node (see diagram 6a, page 32).

6.18 Compare the next *HNP* object in the array and continue setting counters. When all links have been checked, go to the next node in the tree (6.19).

6.19 Repeat for the next node in the tree. When all nodes have been checked, go to 6.20.

[The previous steps set numerous counters for impact and importance values for each tree node point. Steps 6.20-6.34 determine which metrics values are actually required for display, and collect the results from all the nodes into a single set of values for use in the final metrics calculations.]

6.20 *Loop:* Perform the following steps for every element in the array of *NNP* objects from 6.2 or 6.4.

6.21 Get the next element in the array.

6.22-6.23 If the user requested the Clarity metric, increment an aggregate *Clarity* counter by the value for the current node:

Clarity = *Clarity* + (*Clarity* for this node)

6.24-6.25 If the user requested the Involvement metric, increment an aggregate *Links* counter by the number of high-importance links for the current node:

Links = *Links* + (*GT3* for this node)

6.26-6.27 If the user requested the Leverage metric, increment a set of aggregate *Links* counters by the *Importance* counters for the current node:

Links1 = *Links1* + *Importance1*

Links2 = *Links2* + *Importance2*

Links3 = *Links3* + *Importance3*

Links4 = *Links4* + *Importance4*

Links5 = *Links5* + *Importance5*

6.28-6.29 If the user requested the Priority metric, increment a set of aggregate *Links* counters by the *Impact* counters for the current node, then calculate a weighted grand total:

Links1 = *Links1* + *Impact1*

Links2 = *Links2* + *Impact2*

Links3 = *Links3* + *Impact3*

Links4 = Links4 + Impact4

Links5 = Links5 + Impact5

Links = Links1 + 2(Links2) + 3(Links3) + 4(Links4) + 5(Links5)

6.30-6.31 If the user requested the Relative Priority metric, for each of the 20 defined issues, increment a set of aggregate *Links* counters by the *Impact* counters for the current node. Then calculate a weighted grand total of relative links:

For all 20 Issues:

Links1 = Links1 + Impact1

Links2 = Links2 + Impact2

Links3 = Links3 + Impact3

Links4 = Links4 + Impact4

Links5 = Links5 + Impact5

End of loop:

Relative Links = Links1 + 2(Links2) + 3(Links3) + 4(Links4) + 5(Links5)

6.32-6.33 If the user requested the Integration metric, increment a set of aggregate *Links* counters by the *Integration Importance* counters for the current node. Then increment a counter N_1 or N_2 according to whether the node is in unit 1 or unit 2, as stored in 6.15.29:

Links1 = Links1 + Integration Importance1

Links2 = Links2 + Integration Importance2

Links3 = Links3 + Integration Importance3

Links4 = Links4 + Integration Importance4

Links5 = Links5 + Integration Importance5

Increment N_1 if the node is in unit 1

Increment N_2 if the node is in unit 2

6.34 Go to the next tree node point in the array. When all node points have been processed, go to 6.35.

[Steps 6.35-6.46 perform the final metrics calculations.]

6.35-6.36 If the user requested the Clarity metric, apply the following formula to the *Clarity* counter from 6.23:

Clarity = *Clarity* / *N*

where: *N* = size of *NNP* array

6.37-6.38 If the user requested the Involvement metric, apply the following formula to the *Links* counter from 6.25:

$$Involvement = \frac{Links}{N(2^{N-1}-1)}$$

where: *N* = size of *NNP* array

6.39-6.40 If the user requested the Leverage metric, aggregate the *Links1 – Links5* counters from 6.27, then apply the formula to the aggregated value as follows:

Links = *Links1* + 2(*Links2*) + 3(*Links3*) + 4(*Links4*) + 5(*Links5*)

$$Leverage = \frac{Links/2}{5N(2^{N-1}-1)}$$

where: *N* = size of *NNP* array

6.41-6.42 If the user requested the Priority metric, apply the following formula to the *Links* counter from 6.29:

$$Priority = \frac{Links}{10N(2^{N-1}-1)}$$

where: *N* = size of *NNP* array

6.43-6.44 If the user requested the Relative Priority metric, apply the following formulas to the *Links* counters from 6.29 and the *Relative Links* counter from 6.31, then make the Relative Priority calculation:

Priority = *Links* / 10*N*

Relative Priority = *Relative Links* / $10N(2^{N-1}-1)$

Relative Priority = *Priority* / *Relative Priority*

where: *N* = size of *NNP* array

6.45-6.46 If the user requested the Integration metric, aggregate the *Links1 – Links5* counters from 6.33, then apply the formula to the aggregated value and the N_1 and N_2 counters as follows:

$$Links = Links1 + 2(Links2) + 3(Links3) + 4(Links4) + 5(Links5)$$

$$Integration = \frac{Links/2}{5N_1N_2}$$

After calculating all required metrics values, the program sends the data to the display engine for screen display. The display engine is a separate program module, not included in the present discussion (see *3.0 Display options,* page 52).

2.6.6.1 Diagram 6a: Calculating metrics parameters

In order to calculate the various metrics values for the node being traversed, this procedure sets a number of counters which will be used in the final calculations. As indicated in diagram 6, it may be invoked from either step 6.15 (if the node being traversed matches the left side of the link) or step 6.17 (if the node matches the right side). In either case, the values for *Importance* and *Impact* are taken from the corresponding *DIE* object.

6.15.1-6.15.2 Retrieve the left and right *DIE* objects from the *HNP.*

6.15.3-6.15.4 The formula for Involvement defines a highly important link as one with an assigned overall importance value of greater than or equal to 3. If the stored *Importance* value in either the left or right *DIE* is in this range, increment a counter *GT3* for this node.

[The Leverage metric sums a person's confirmed links, weighted by their importance values. Steps 6.15.5-6.15.15 set counters to record the numbers of links at each importance level from 1 through 5.]

6.15.5 Set the value of *Importance* to the lesser of the values in the left and right *DIE*s.

6.15.6-6.15.7 If the *Importance* value in the *DIE* is 1, increment a counter *Importance 1* for this node by the value of *Importance.*

6.15.8-6.15.9 If the *Importance* value in the *DIE* is 2, increment a counter *Importance 2* for this node by the value of *Importance.*

6.15.10-6.15.11 If the *Importance* value in the *DIE* is 3, increment a counter *Importance 3* for this node by the value of *Importance.*

6.15.12-6.15.13 If the *Importance* value in the *DIE* is 4, increment a counter *Importance 4* for this node by the value of *Importance.*

6.15.14-6.15.15 If the *Importance* value in the *DIE* is 5, increment a counter *Importance 5* for this node by the value of *Importance.*

[The Priority metric sums links weighted by impact values. Steps 6.15.16-6.15.27 set counters to record the numbers of half-links at each impact level from 1 through 5, for all issues included in the analysis definition.]

6.15.16 Perform the following steps for each element in the array:

6.15.17-6.15.18 If the *Impact* value in the *DIE* on this issue is 1, increment a counter *Impact 1* for this node.

6.15.19-6.15.20 If the *Impact* value in the *DIE* on this issue is 2, increment a counter *Impact 2* for this node.

6.15.21-6.15.22 If the *Impact* value in the *DIE* on this issue is 3, increment a counter *Impact 3* for this node.

6.15.23-6.15.24 If the *Impact* value in the *DIE* on this issue is 4, increment a counter *Impact 4* for this node.

6.15.25-6.15.26 If the *Impact* value in the *DIE* on this issue is 5, increment a counter *Impact 5* for this node.

6.15.27 Go to the next defined issue. When impact values for all issues have been processed, go to 6.15.28.

[The Integration metric sums the number of interconnecting links between two separate organizational units, weighted by the importance of the links. The remaining steps isolate those links that connect different units and set counters to record the numbers of links at each overall *Importance* value from 1 through 5.]

6.15.28 Compare the values of *Left person unit number* and *Right person unit number* in the *HNP* to determine whether the left side and the right side of the *HNP* are in different organizational units. If they are, proceed to 6.15.29. Otherwise, exit.

6.15.29 Record whether the node is in unit 1 or unit 2.

6.15.30 Set the value of *Importance* to the lesser of the values in the left and right *DIE*s.

6.15.31-6.15.32 If the *Importance* value in the *DIE* is 1, increment a counter *Integration Importance 1* for this node by the value of *Importance.*

6.15.33-6.15.34 If the *Importance* value in the *DIE* is 2, increment a counter *Integration Importance 2* for this node by the value of *Importance.*

6.15.35-6.15.36 If the Importance value in the DIE is 3, increment a counter Integration Importance 3 for this node by the value of Importance.

6.15.37-6.15.38 If the *Importance* value in the *DIE* is 4, increment a counter *Integration Importance 4* for this node by the value of *Importance.*

6.15.39-6.15.40 If the *Importance* value in the *DIE* is 5, increment a counter *Integration Importance 5* for this node by the value of *Importance.*

After setting all counters, return to the higher-level process at 6.18.

3.0 DISPLAY OPTIONS

The metrics analysis produces data consisting of the results of the various metrics calculations, which is stored in a database. Options for retrieving this information and displaying it fall into two main categories:

1. using a prepackaged report generator
2. creating a custom display application

3.1 REPORT GENERATORS

Numerous report-writing packages exist for the purpose of processing database queries and retrieving specified data. This approach requires minimal programming. Its disadvantage is that the capabilities of such packages are fixed and limited: for example, they typically produce only tabular reports and cannot display data in a graphic, hierarchical format.

Representative applications in this category are report writers such as Crystal Reports (Crystal Decisions company), those included in standard database packages from vendors such as Oracle, Sybase, IBM, and Microsoft, and numerous independent and shareware ODBC-compliant database viewers.

3.2 CUSTOM DISPLAY APPLICATIONS

The more useful way of processing the output data is to display them in graphical form, superimposing the connections against a hierarchical tree structure of persons or issues. This approach requires the creation of an application that can read from the database and generate instructions for creating a graphical tree structure. These instructions are then passed to a display-engine application programming interface (API). The API may be capable of rendering hierarchical output in either a two-dimensional or a three-dimensional display. The EnCompass system itself uses an API created for the purpose by the Parasol Development company (see *3.2.4 The Parasol application and display engine*, page 54).

3.2.1 Hierarchical data display libraries

The basic way of implementing the hierarchical display is to use a standard API or class library providing two-dimensional tree-drawing capabilities. Examples of such libraries are the following (names of developers or vendors are in parentheses):

WXP3D (The Whole Experience)

Orion3D (Gabriel Peyré)

Parasol Meta Classes (Parasol Development)

Legus3D (Branimir Karadzic)

Plush (Nullsoft, Inc.)

blaxxunContact (blaxxun interactive)

Architect III (Geometric Computing)

E3d Engine (Act 3d Interactive)

ATV tree library (Christian Zmasek)

Visual C++ (Microsoft)

Swing Jtree (Java class)

tree.hh tree class for Borland C++ (Kasper Peeters)

GTK+ (GNU PL)

PopChart (Corda)

3.2.2 3D display engines

A fully enhanced custom display application provides the ability to represent the relationships in a three-dimensional display. For this there are 3D analogues to the APIs already mentioned, including the following:

Microsoft Windows:

OpenGL (multiple vendors)

Visual C++ (Microsoft)

Direct3D (Microsoft)

Parasol Meta Classes (Parasol Development)

Apple Macintosh:

QuickDraw3D (Apple)

UNIX/Linux:

Ploticus (Seawall Group Productions)

Multiple platforms:

Quesa (Open Source)

3.2.3 Graph libraries

A further enhancement of the graphic output display makes use of a graphing API. Representative examples include:

Boost Graph Library (Lee, Lumsdaine, Siek)

LEDA (Max Planck Institut)

Graph Template Library (GTL) (University of Passau)

Stanford GraphBase (Knuth, Addison Wesley)

Graph Layout Toolkit (Tom Sawyer Software)

Triangulated Surface Library (GNU open source)

C+O Class Library Foundation Data Structures (Parasol Development)

3.2.4 The Parasol application and display engine

The custom application that provides data display and other support functions to the EnCompass system is provided by Parasol Development. The Parasol application consists of two main components (see figure 13, page 56).

The *class library* provides low- and mid-level functions for handling data structures, window objects and behaviors, application interface, and basic database access. This component is documented in the volumes *C+Objects: Volume One: Foundation Data Structures* and *C+O Class Library Foundation Data Structures: User's Guide* and *Reference Manual.*

At a higher abstraction level, the Parasol *meta-application* includes advanced database access, Parasol-specific user interface functions, and floating-point functions which provide the 3D calculation and transformation routines making up the display engine. Documentation for this component appears in the *Parasol Developer User Guide* .

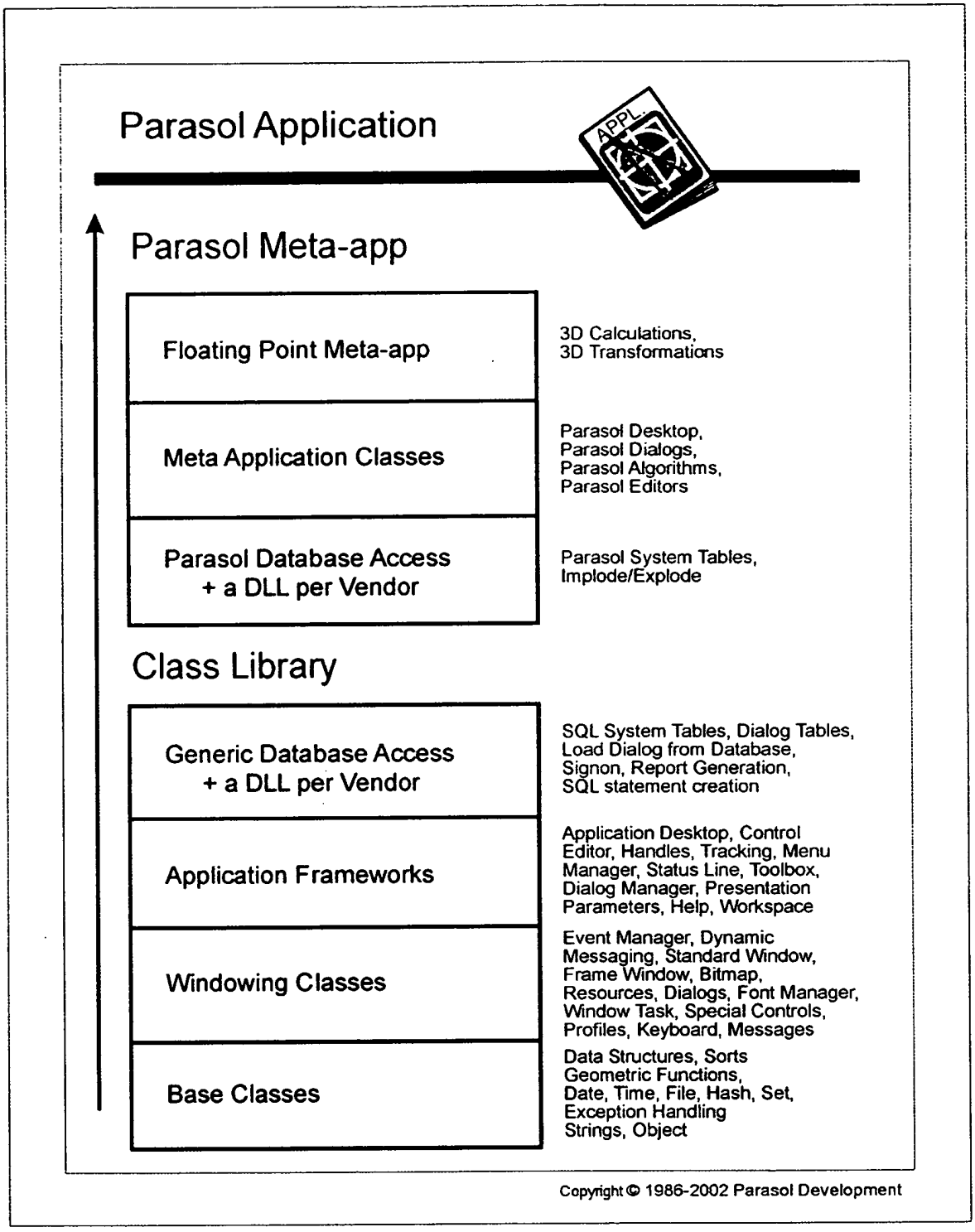

Figure 13: Parasol application structure

4.0 METRICS DISPLAY EXAMPLE

As an aid to understanding the concepts employed above, the following is a step-by-step example of the procedure for displaying metrics figures in the EnCompass application. To follow the demonstration, install and run the EnCompass application as directed in the file README.TXT, on the installation CD/ROM.

To view metrics figures:

Display the organization structure:

1. On the vertical tool bar at the left side of the application main window, click the "Person" icon (or choose *General Data : Person* from the *File* menu).

2. In the *Search Person* dialog, select "Sample DB" from the drop-down list in the *Project* field. In the *Corporate Title* field, select "President."

3. Click **Search.** The program locates the organization's president and CEO, "Judith Ball." Select (highlight) this name in the list.

4. From the *Search* menu, choose *Explode Down All.* EnCompass displays the organization structure from this node.

   (If the organization structure does not appear, make sure that the "All" box is checked, and the number of levels is set to at least 6. If necessary, change the settings and click **Redo current query** to redraw the display.)

5. From the *View* menu, choose *Outline.* EnCompass displays a two-dimensional view of the organization structure.

Load an analysis:

1. On the vertical tool bar at the left side of the application main window, click the "Analysis" icon (or choose *Analysis* from the *File* menu).

2. In the *Search Analysis* dialog, select "Sample DB" from the drop-down list in the *Project* field. Click **Search.** The program displays a list of analysis definitions.

3. In the *Analysis List,* select item AN0000025, "Everything." From the *Search* menu, choose *Load Analysis.* EnCompass displays the organization structure from this node.

4. From the *View* menu, choose *Outline.* EnCompass displays a two-dimensional view of the organization structure.

Set display layout to show metrics:

1. From the *Options* menu, choose *Display Layout.*

2. In the *Display Layout* dialog, select the *Private Design* option button. In the *Layout Table* field, make sure *Person* is selected.

3. From the *Layout Person* drop-down list, select "Node Clarity."

4. Click **Set** in the field *2. Append Column to Outline.* In the *Separator* field to the right, enter ";" and a space.

5. Repeat steps 3 and 4, placing "Immed. Clarity" in column 3 and "Ext. Clarity" in column 4. The dialog should appear as in figure 14.

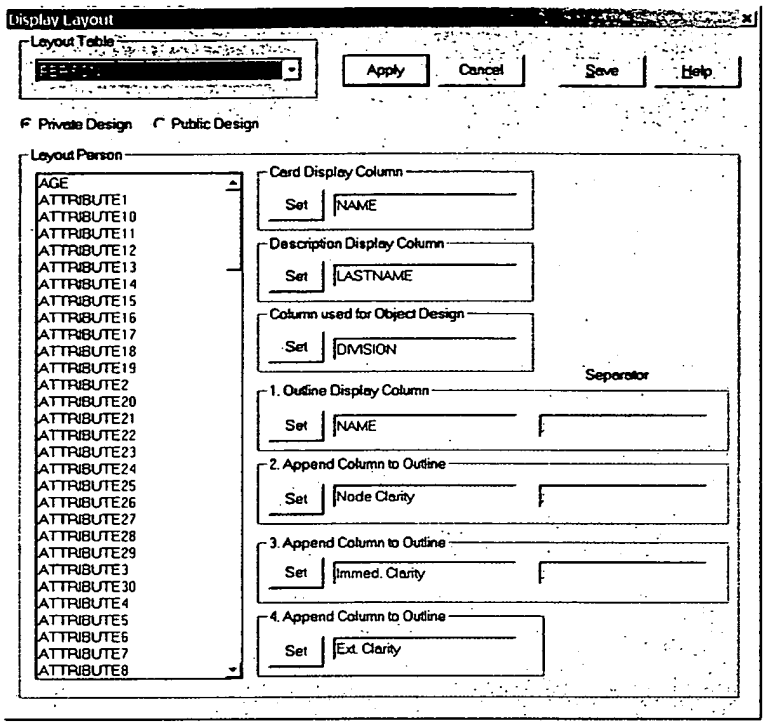

Figure 14: Metrics display layout

6. Click **Apply.**

7. In the *Organization View* window, click **Redo current query** to redraw the display.

8. From the *Show* menu, choose *Data Collection Links* to apply the data to the displayed structure. The *Organization View* window display should appear as in figure 15.

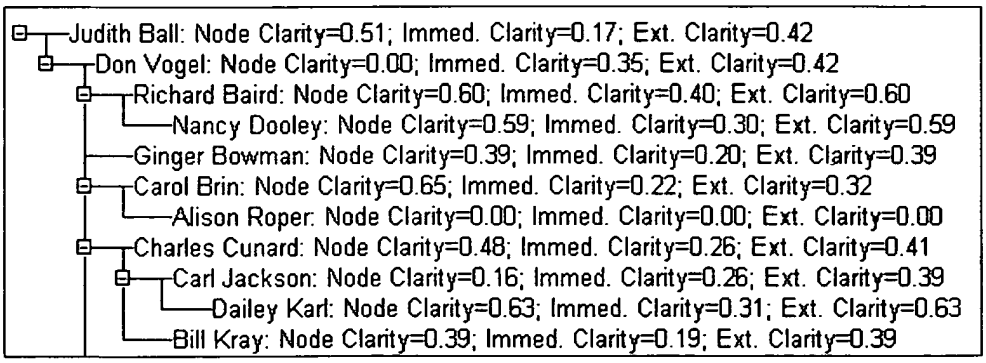

Figure 15: Displaying metrics results